

TUGAS AKHIR - KI1502

PENGEMBANGAN APLIKASI PENDUKUNG TERAPI FOBIA SERANGGA MENGGUNAKAN TEKNOLOGI AUGMENTED REALITY

M IRFAN RAMADHAN
NRP 5111100148

Dosen Pembimbing I
Ridho Rahman H., S.Kom., M.Sc.

Dosen Pembimbing II
Dr.Eng. Darlis Herumurti, S.Kom., M.Kom.

JURUSAN TEKNIK INFORMATIKA
Fakultas Teknologi Informasi
Institut Teknologi Sepuluh Nopember
Surabaya 2017



TUGAS AKHIR - KI1502

**PENGEMBANGAN APLIKASI PENDUKUNG TERAPI
FOBIA SERANGGA MENGGUNAKAN TEKNOLOGI
AUGMENTED REALITY**

**M IRFAN RAMADHAN
NRP 5111100148**

**Dosen Pembimbing I
Ridho Rahman H., S.Kom., M.Sc.**

**Dosen Pembimbing II
Dr.Eng. Darlis Herumurti, S.Kom., M.Kom.**

**JURUSAN TEKNIK INFORMATIKA
Fakultas Teknologi Informasi
Institut Teknologi Sepuluh Nopember
Surabaya 2017**

(Halaman ini sengaja dikosongkan)



UNDERGRADUATE THESIS - KI1502

**APPLICATION DEVELOPMENT FOR SUPPORTING
INSECT PHOBIA THERAPY USING AUGMENTED
REALITY TECHNOLOGY**

**M IRFAN RAMADHAN
NRP 5111100148**

**Supervisor I
Ridho Rahman H., S.Kom., M.Sc.**

**Supervisor II
Dr.Eng. Darlis Herumurti, S.Kom., M.Kom.**

**DEPARTMENT OF INFORMATICS
Faculty of Information Technology
Institut Teknologi Sepuluh Nopember
Surabaya 2017**

(Halaman ini sengaja dikosongkan)

LEMBAR PENGESAHAN

PENGEMBANGAN APLIKASI PENDUKUNG TERAPI FOBIA SERANGGA MENGGUNAKAN TEKNOLOGI AUGMENTED REALITY

TUGAS AKHIR

Diajukan Untuk Memenuhi Salah Satu Syarat
Memperoleh Gelar Sarjana Komputer
pada

Bidang Studi Dasar dan Terapan Komputasi
Program Studi S-1 Jurusan Teknik Informatika
Fakultas Teknologi Informasi
Institut Teknologi Sepuluh Nopember

Oleh

M IRFAN RAMADHAN

NRP : 5111100148

Disetujui oleh Dosen Pembimbing Tugas Akhir

1. Ridho Rahman H., S.Kom., M.Sc.
NIP: 198702132014041001 (Pembimbing 1)
2. Dr.Eng. Darlis Herumurti, S.Kom., M.Kom.
NIP: 197712172003121001 (Pembimbing 2)



SURABAYA

JULI, 2017

(Halaman ini sengaja dikosongkan)

PENGEMBANGAN APLIKASI PENDUKUNG TERAPI FOBIA SERANGGA MENGGUNAKAN TEKNOLOGI AUGMENTED REALITY

Nama : M Irfan Ramadhan
NRP : 5111100148
Jurusan : Teknik Informatika
Fakultas Teknologi Informasi ITS
Dosen Pembimbing I : Ridho Rahman H., S.Kom., M.Sc.
Dosen Pembimbing II : Dr.Eng. Darlis Herumurti, S.Kom.,
M.Kom.

ABSTRAK

Aplikasi pendukung terapi fobia serangga merupakan aplikasi yang berjalan di perangkat Android yang digunakan untuk membantu proses terapi bagi penderita fobia serangga. Tugas akhir ini berfokus pada pembuatan aplikasi sebagai media simulasi proses terapi fobia serangga menggunakan teknologi augmented reality.

Metode yang digunakan dalam tugas akhir ini adalah dengan menampilkan serangga virtual dimana pengguna bisa melihat dan berinteraksi dengan cara memukul serangga tersebut. Serangga virtual dapat ditampilkan berdasarkan 4 parameter utama yang terdiri dari jenis serangga, jumlah serangga, pergerakan serangga, dan ukuran serangga. Terdapat juga 1 parameter tambahan yang dapat diaktifkan agar pengguna bisa membunuh serangga virtual dengan cara memukulnya. Parameter tersebut dapat disesuaikan sesuai prosedur terapi yang akan dilakukan. Selain itu aplikasi juga mampu menyimpan skenario terapi sehingga pengguna dapat melakukan terapi secara mandiri.

Hasil uji coba aplikasi yang dilakukan menunjukkan bahwa aplikasi ini efektif digunakan sebagai pendukung terapi fobia serangga.

Kata kunci: Augmented Reality, Terapi Fobia Serangga.

APPLICATION DEVELOPMENT FOR SUPPORTING INSECT PHOBIA THERAPY USING AUGMENTED REALITY TECHNOLOGY

Name : M Irfan Ramadhan
NRP : 5111100148
Department : Department of Informatics
Faculty of Information Technology ITS
Supervisor I : Ridho Rahman H., S.Kom., M.Sc.
Supervisor II : Dr.Eng. Darlis Herumurti, S.Kom.,
M.Kom.

ABSTRACT

Application for supporting insect phobia therapy is an application that runs on Android device to support therapy for insect phobia sufferers.

The method used in this undergraduate thesis is showing virtual insect which the user can see and interact with them by hitting them. The virtual insect can be shown depends on 4 main parameter consist of insect type, insect amount, insect movement, and insect size. There is also 1 extra parameter that can be activated so the user can kill the virtual insect by hitting them. Those parameter can be adjusted to fit with the therapy procedure which will be done. Futhermore, the app can save therapy scenario so users can do self therapy.

The test result on this application shows that this application is effective to support insect therapy phobia.

Keywords: Augmented Reality, Insect Phobia Therapy.

KATA PENGANTAR

Puji syukur penulis kehadiran Tuhan Yang Maha Esa karena berkat rahmat dan karunia-Nya penulis dapat menyelesaikan Tugas Akhir yang berjudul

PENGEMBANGAN APLIKASI PENDUKUNG TERAPI FOBIA SERANGA MENGGUNAKAN TEKNOLOGI AUGMENTED REALITY

Tugas Akhir ini merupakan salah satu syarat untuk memperoleh gelar Sarjana Komputer di Jurusan Teknik Informatika, Fakultas Teknologi Informasi, Institut Teknologi Sepuluh Nopember Surabaya.

Penulis ingin menyampaikan terima kasih yang sebesar-besarnya atas dukungan dan semangat yang diberikan dan membantu penulis baik secara langsung maupun tidak dalam menyelesaikan Tugas Akhir ini. Penulis ingin mengucapkan terima kasih kepada:

1. Tuhan Yang Maha Esa karena berkat rahmat dan karunianya penulis berhasil menyelesaikan Tugas Akhir dengan baik.
2. Orang tua dan keluarga penulis yang selalu memberikan doa dan dukungan selama penulis kuliah di Jurusan Teknik Informatika ITS.
3. Bapak Dr.Eng. Darlis Herumurti, S.Kom., M.Kom. selaku Ketua Jurusan Teknik Informatika Fakultas Teknologi Informasi Institut Teknologi Sepuluh Nopember dan Dosen Pembimbing II Tugas Akhir yang telah memberikan banyak ilmu selama penulis berkuliah di Jurusan Teknik Informatika ITS serta seluruh dukungan dan bantuan selama penulis menyelesaikan Tugas Akhir.
4. Bapak Radityo Anggoro, S.Kom., M.Sc. selaku Koordinator Tugas Akhir.
5. Bapak Ridho Rahman H., S.Kom., M.Sc. selaku Dosen Pembimbing I Tugas Akhir yang telah memberikan

bimbingan dan dukungan selama penulis menyelesaikan Tugas Akhir.

6. Bapak dan Ibu Dosen Jurusan Teknik Informatika ITS yang telah memberikan ilmu selama penulis berkuliah di Jurusan Teknik Informatika ITS.
7. Seluruh staf dan karyawan Jurusan Teknik Informatika ITS yang telah membantu selama penulis berkuliah di Jurusan Teknik Informatika ITS.
8. Teman-teman seangkatan yang selalu siap sedia ketika penulis mengalami kesulitan.
9. Teman-teman angkatan 2009, 2010, 2012, 2013 yang sudah memberikan pengalaman selama kuliah di Teknik Informatika ini.
10. Seluruh pihak lain yang tidak bisa saya sebutkan satu persatu yang telah mendukung penulis.

Penulis mohon maaf apabila terdapat kekurangan dalam penulisan Tugas Akhir ini. Kritik dan saran penulis harapkan untuk perbaikan dan pembelajaran di kemudian hari. Semoga Tugas Akhir ini dapat memberikan manfaat yang sebesar-besarnya.

Surabaya, Juli 2017

Penulis

DAFTAR ISI

LEMBAR PENGESAHAN.....	Error! Bookmark not defined.
ABSTRAK	ix
ABSTRACT	x
KATA PENGANTAR.....	xi
DAFTAR ISI	xiii
DAFTAR GAMBAR	xix
DAFTAR TABEL	xxv
BAB I PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	2
1.3 Batasan Masalah.....	2
1.4 Tujuan.....	2
1.5 Manfaat.....	3
1.6 Metodologi	3
1.7 Sistematika Penulisan.....	4
BAB II TINJAUAN PUSTAKA	7
2.1 Fobia Serangga	7
2.2 Terapi Fobia Serangga.....	8
2.3 <i>Augmented Reality</i>	9
2.4 SDK.....	9
2.5 Unity	9
2.6 Vuforia	10
2.7 Finite-State Machine	10
BAB III ANALISIS DAN PERANCANGAN.....	13

3.1	Analisis Sistem	13
3.2	Perancangan Model 3D	14
3.3	Perancangan Perilaku Serangga.....	14
3.3.1	Rancangan Perilaku Menambah Serangga	14
3.3.2	Rancangan Perilaku Serangga Berjalan.....	14
3.3.3	Rancangan Perilaku Membunuh Serangga.....	15
3.4	Perancangan Algoritma	16
3.4.1	Rancangan Algoritma Penambahan Serangga.....	16
3.4.2	Rancangan Algoritma Pengurangan Serangga	17
3.3.4	Algoritma Menjalankan Serangga	17
3.3.5	Algoritma Menghentikan Serangga.....	18
3.3.6	Algoritma Pertumbukkan Serangga.....	18
3.3.7	Algoritma Pembatasan Pergerakan Serangga.....	19
3.3.8	Algoritma Pembesaran Serangga.....	19
3.3.9	Algoritma Pengecilan Serangga	20
3.3.10	Rancangan Algoritma Bunuh Serangga.....	20
3.3.11	Rancangan Algoritma Mengembalikan Posisi Serangga Pertama	21
3.3.12	Rancangan Algoritma Penambahan Skenario	22
3.3.13	Rancangan Algoritma Pengubahan Skenario	22
3.3.14	Rancangan Algoritma Penghapusan Skenario.....	23
3.3.15	Rancangan Algoritma Menjalankan Skenario	23
3.3.16	Rancangan Algoritma Menghentikan Skenario....	24
3.5	Perancangan Perangkat Lunak.....	24
3.5.1	Deskripsi Umum Perangkat Lunak.....	25
3.5.2	Spesifikasi Kebutuhan Fungsional	26

3.5.3	Spesifikasi Kebutuhan Non-Fungsional	26
3.5.4	Karakteristik Pengguna	26
3.6	Perancangan Sistem.....	27
3.6.1	Perancangan Diagram Kasus Penggunaan	27
3.6.2	Perancangan Skenario Kasus Penggunaan	31
BAB IV IMPLEMENTASI.....		51
4.1	Lingkungan Implementasi.....	51
4.2	Implementasi Menambah Serangga	52
4.2.1	Implementasi Pembuatan Scene dan <i>Load</i> Objek.....	52
4.2.2	Implementasi Pembuatan <i>Script</i>	55
4.2.3	Implementasi Pembuatan <i>Canvas</i> , <i>Button</i> , dan <i>Text</i>	56
4.4.4	Implementasi Penambahan Serangga	59
4.2	Implementasi Mengurangi Serangga	61
4.3	Implementasi Menjalankan dan Menghentikan Serangga.....	61
4.5.1	Implementasi Serangga Berjalan dan Serangga Diam	62
4.5.2	Implementasi Sistem <i>Collision</i>	63
4.4	Implementasi Batas Pergerakan Serangga.....	65
4.5	Implementasi Memperbesar Serangga	65
4.6	Implementasi Memperkecil Serangga	66
4.7	Implementasi Fitur Bunuh Serangga	67
4.7.1	Implementasi <i>Virtual Button</i>	67
4.7.2	Implementasi Pengaktifan dan Penonaktifan Fitur Bunuh Serangga	72
4.7.3	Implementasi <i>Respawn</i> Serangga	73

4.8	Implementasi Fitur Mengembalikan Posisi Serangga Pertama	75
4.9	Implementasi Menambah Skenario	75
4.10	Implementasi Mengubah Skenario	79
4.11	Implementasi Menghapus Skenario.....	81
4.12	Implementasi Menjalankan Skenario	82
4.13	Implementasi Menghentikan Skenario	85
4.14	Pembuatan Proyek	86
4.15	Cara Menjalankan Aplikasi	87
BAB V PENGUJIAN DAN EVALUASI		89
5.1	Lingkungan Uji Coba	89
5.1.1	Lingkungan Perangkat Komputer.....	89
5.1.2	Lingkungan Perangkat Android.....	90
5.2	Pengujian Fungsionalitas.....	90
5.2.1	Cara Menjalankan Aplikasi	90
5.2.2	Skenario Uji Coba Fungsionalitas	91
5.2.3	Hasil Uji Coba	91
5.3	Pengujian Non-Fungsionalitas.....	139
5.3.1	Skenario Uji Coba Non-Fungsionalitas	139
5.3.2	Hasil Uji Coba	140
5.4	Evaluasi	146
5.4.1	Evaluasi Pengujian Fungsionalitas	146
5.4.2	Evaluasi Pengujian Non-Fungsionalitas	148
BAB VI KESIMPULAN.....		149
6.1.	Kesimpulan.....	149
6.2.	Saran.....	149

DAFTAR PUSTAKA.....	151
LAMPIRAN.....	153
BIODATA PENULIS.....	201

(Halaman ini sengaja dikosongkan)

DAFTAR GAMBAR

Gambar 2.1 Contoh diagram <i>state</i> sederhana.....	11
Gambar 3.1 FSM menambah serangga	14
Gambar 3.2 FSM srangga berjalan.....	15
Gambar 3.3 FSM membunuh serangga.....	15
Gambar 3.4 <i>Pseudocode</i> algoritma penambahan serangga petama	16
Gambar 3.5 <i>Pseudocode</i> algoritma penambahan serangga kedua hingga kesepuluh.....	17
Gambar 3.6 <i>Pseudocode</i> algoritma pengurangan serangga.....	17
Gambar 3.7 <i>Pseudocode</i> algoritma menjalankan serangga.....	17
Gambar 3.8 <i>Pseudocode</i> algoritma menghentikan serangga.....	18
Gambar 3.9 <i>Pseudocode</i> algoritma pertumbukkan serangga	18
Gambar 3.10 <i>Pseudocode</i> algoritma pembatasan pergerakan serangga.....	19
Gambar 3.11 <i>Pseudocode</i> algoritma perbesaran serangga	19
Gambar 3.12 <i>Pseudocode</i> algoritma pengecilan serangga	20
Gambar 3.13 <i>Pseudocode</i> algoritma bunuh serangga	21
Gambar 3.14 <i>Pseudocode</i> algoritma mengembalikan posisi serangga pertama	21
Gambar 3.15 <i>Pseudocode</i> algoritma penambahan skenario.....	22
Gambar 3.16 <i>Pseudocode</i> algoritma pengubahan skenario.....	23
Gambar 3.17 <i>Pseudocode</i> algoritma penambahan skenario.....	23
Gambar 3.18 <i>Pseudocode</i> algoritma menjalankan skenario.....	24
Gambar 3.19 <i>Pseudocode</i> algoritma menjalankan skenario.....	24
Gambar 3.20 Diagram arsitektur sistem.....	25
Gambar 3.21 Diagram kasus penggunaan	28
Gambar 4.1 Penambahan <i>scene</i>	52
Gambar 4.2 Tampilan <i>scene</i> yang telah dibuat	53
Gambar 4.3 Tampilan <i>project explorer</i>	53
Gambar 4.4 Inseri objek ke dalam proyek	54
Gambar 4.5 Properti objek	54
Gambar 4.6 Pembuatan <i>script</i> (1).....	55
Gambar 4.7 Pembuatan <i>script</i> (2).....	55

Gambar 4.8 Pembuatan <i>script</i> (3).....	56
Gambar 4.9 Pembuatan <i>button</i> dan <i>text</i>	57
Gambar 4.10 Tampilan <i>button</i> dan <i>text</i> yang sudah dibuat	57
Gambar 4.11 Penambahan objek <i>game</i> pada fungsi <i>on click</i>	58
Gambar 4.12 Penambahan fungsi pada fungsi <i>on click</i>	58
Gambar 4.13 Impor <i>package</i> Vuforia.....	59
Gambar 4.14 <i>Script</i> menambah serangga	61
Gambar 4.15 <i>Script</i> mengurangi serangga	61
Gambar 4.16 <i>Script</i> moveInsect	62
Gambar 4.17 <i>Script</i> menjalankan serangga	62
Gambar 4.18 <i>Script</i> menghentikan serangga	63
Gambar 4.19 Penambahan <i>Collider</i> dan <i>Rigidbody</i>	64
Gambar 4.20 Penambahan <i>Collider</i> dan <i>Rigidbody</i>	64
Gambar 4.21 <i>Script Collision</i>	65
Gambar 4.22 Kolom “ <i>Is Trigger</i> ”	65
Gambar 4.23 <i>Script</i> pembatasan gerakan serangga	65
Gambar 4.24 <i>Script</i> memperbesar serangga	66
Gambar 4.25 <i>Script</i> memperkecil serangga.....	67
Gambar 4.26 Penambahan <i>virtual button</i>	68
Gambar 4.27 Penambahan <i>virtual button</i> ke objek <i>game</i>	69
Gambar 4.28 <i>Virtual button</i> yang telah berhasil ditambahkan	69
Gambar 4.29 Penambahan <i>script VEventHandler</i>	70
Gambar 4.30 <i>Script VEventHandler</i> saat <i>virtual button</i> ditekan	71
Gambar 4.31 <i>Script VEventHandler</i> saat <i>virtual button</i> dilepas	71
Gambar 4.32 <i>Script VEventHandler</i> saat <i>virtual button</i> dilepas (2)	72
Gambar 4.33 <i>Script</i> mengaktifkan fitur bunuh serangga.....	72
Gambar 4.34 <i>Script</i> menonaktifkan fitur bunuh serangga.....	73
Gambar 4.35 <i>Script respawn</i>	74
Gambar 4.36 <i>Script</i> mengembalikan posisi serangga pertama	75
Gambar 4.37 <i>Script</i> menambah skenario.....	79
Gambar 4.38 <i>Script</i> mengubah skenario.....	81
Gambar 4.39 <i>Script</i> menghapus skenario	82
Gambar 4.40 <i>Script</i> menjalankan skenario.....	85

Gambar 4.41 <i>Script</i> menghentikan skenario	85
Gambar 4.42 Pembuatan proyek	86
Gambar 4.43 Pembuatan proyek (2).....	87
Gambar 5.1 Tampilan tombol tambah serangga pada menu pengaturan	94
Gambar 5.2 Tampilan pilihan serangga yang dapat ditambahkan	95
Gambar 5.3 Tampilan saat menambahkan serangga	95
Gambar 5.4 Hasil penambahan 1 serangga pada menu pengaturan	96
Gambar 5.5 Hasil penambahan 1 serangga pada tampilan utama	96
Gambar 5.6 Tampilan saat menambahkan banyak serangga.....	97
Gambar 5.7 Hasil penambahan banyak serangga pada menu pengaturan	97
Gambar 5.8 Hasil penambahan banyak serangga pada tampilan utama	98
Gambar 5.9 Tampilan sebelum serangga dikurangi pada menu pengaturan	100
Gambar 5.10 Tampilan sebelum serangga dikurangi pada tampilan utama	100
Gambar 5.11 Tampilan saat mengurangi 2 serangga	101
Gambar 5.12 Hasil pengurangan 2 serangga pada tampilan utama	101
Gambar 5.13 Tampilan saat menjalankan serangga	104
Gambar 5.14 Hasil menjalankan serangga pada menu pengaturan	104
Gambar 5.15 Hasil menjalankan serangga pada tampilan utama	105
Gambar 5.16 Tampilan saat menghentikan serangga.....	105
Gambar 5.17 Hasil menghentikan serangga pada menu pengaturan	106
Gambar 5.18 Hasil menghentikan serangga pada tampilan utama	106
Gambar 5.19 Tampilan sebelum memperbesar serangga pada menu pengaturan	108

Gambar 5.20 Tampilan sebelum memperbesar serangga pada tampilan utama	109
Gambar 5.21 Tampilan saat memperbesar serangga	109
Gambar 5.22 Hasil memperbesar serangga pada tampilan utama	110
Gambar 5.23 Tampilan sebelum memperkecil serangga pada menu pengaturan	112
Gambar 5.24 Tampilan sebelum memperkecil serangga pada tampilan utama	112
Gambar 5.25 Tampilan saat memperkecil serangga	113
Gambar 5.26 Hasil memperkecil serangga pada tampilan utama	113
Gambar 5.27 Tampilan saat mengaktifkan fitur bunuh serangga	116
Gambar 5.28 Tampilan setelah mengaktifkan fitur bunuh serangga pada menu pengaturan	116
Gambar 5.29 Uji coba bunuh serangga dengan cara dipukul saat berada di dalam marker	117
Gambar 5.30 Tampilan saat menonaktifkan fitur bunuh serangga	117
Gambar 5.31 Tampilan setelah menonaktifkan fitur bunuh serangga	118
Gambar 5.32 Tampilan sebelum mengembalikan posisi serangga pertama	120
Gambar 5.33 Tampilan saat mengembalikan posisi serangga pertama	120
Gambar 5.34 Hasil setelah mengembalikan posisi serangga pertama ke tengah <i>marker</i>	121
Gambar 5.35 Tampilan tombol kelola skenario pada menu terapis	124
Gambar 5.36 Tampilan menu skenario	125
Gambar 5.37 Tampilan menu tambah skenario (1)	125
Gambar 5.38 Tampilan menu tambah skenario (2)	126
Gambar 5.39 Tampilan menu tambah skenario (3)	126

Gambar 5.40 Tampilan tombol ubah skenario pada menu skenario	130
Gambar 5.41 Tampilan pilihan skenario yang dapat diubah.....	130
Gambar 5.42 Tampilan menu ubah skenario.....	131
Gambar 5.43 Tampilan tombol hapus skenario pada menu skenario	133
Gambar 5.44 Tampilan pilihan skenario yang dapat dihapus ...	133
Gambar 5.45 Tampilan setelah menghapus skenario 2	134
Gambar 5.46 Tampilan pilihan skenario yang dapat dijalankan sebagai terapis	136
Gambar 5.47 Tampilan pilihan skenario yang dapat dijalankan sebagai pasien.....	137
Gambar 5.48 Tampilan saat menjalankan skenario sebagai terapis	137
Gambar 5.49 Tampilan saat menjalankan skenario sebagai pasien	138
Gambar 5.50 Tampilan setelah menekan tombol stop skenario sebagai terapis	138
Gambar 5.51 Tampilan setelah menekan tombol stop skenario sebagai pasien.....	139
Gambar 5.52 Uji coba terapi fobia serangga	142
Gambar A.1 <i>Script appmanager</i>	160
Gambar A.2 <i>Script moveInsect</i>	161
Gambar A.3 <i>Script VBEventHandler</i>	168
Gambar A.4 <i>Script scenarioManager</i>	186

(Halaman ini sengaja dikosongkan)

DAFTAR TABEL

Tabel 3.1 Karakteristik pengguna	27
Tabel 3.2 Skenario kasus penggunaan.....	29
Tabel 3.3 Skenario kasus penggunaan menambah serangga	32
Tabel 3.4 Skenario kasus penggunaan mengurangi serangga	33
Tabel 3.5 Skenario kasus penggunaan menjalankan serangga	34
Tabel 3.6 Skenario kasus penggunaan menghentikan serangga...	35
Tabel 3.7 Skenario kasus penggunaan memperbesar serangga...	36
Tabel 3.8 Skenario kasus penggunaan memperkecil serangga.....	37
Tabel 3.9 Skenario kasus penggunaan mengaktifkan fitur bunuh serangga.....	38
Tabel 3.10 Skenario kasus penggunaan menonaktifkan fitur bunuh serangga.....	39
Tabel 3.11 Skenario kasus penggunaan mengembalikan posisi serangga pertama	40
Tabel 3.12 Skenario kasus penggunaan menambah skenario.....	41
Tabel 3.13 Skenario kasus penggunaan mengubah skenario	44
Tabel 3.14 Skenario kasus penggunaan menghapus skenario	47
Tabel 3.15 Skenario kasus penggunaan menjalankan skenario...	48
Tabel 3.16 Skenario kasus penggunaan menghentikan skenario	49
Tabel 4.1 Lingkungan implementasi perangkat lunak	51
Tabel 5.1 Lingkungan Perangkat Komputer	89
Tabel 5.2 Lingkungan Perangkat Android	90
Tabel 5.3 Hasil uji coba menambah serangga	92
Tabel 5.4 Hasil uji coba mengurangi serangga	99
Tabel 5.5 Hasil uji coba menjalankan dan menghentikan serangga	102
Tabel 5.6 Hasil uji coba memperbesar serangga	107
Tabel 5.7 Hasil uji coba memperbesar serangga	111
Tabel 5.8 Hasil uji coba mengaktifkan dan menonaktifkan fitur bunuh serangga.....	114
Tabel 5.9 Hasil uji coba mengembalikan posisi serangga pertama	119
Tabel 5.10 Hasil uji coba menambahkan skenario	122

Tabel 5.11 Hasil uji coba mengubah skenario.....	127
Tabel 5.12 Hasil uji coba menghapus skenario	132
Tabel 5.13 Hasil uji coba menjalankan skenario.....	134
Tabel 5.14 Daftar nama penguji coba aplikasi	140
Tabel 5.15 Rata-rata hasil uji coba performa aplikasi	142
Tabel 5.16 Rata-rata hasil uji coba <i>immersive</i>	144
Tabel 5.17 Rata-rata hasil uji coba keefektifan aplikasi.....	145
Tabel 5.18 Rekapitulasi hasil uji coba fungsionalitas	147
Tabel 5.19 Rekapitulasi hasil uji coba fungsionalitas	148
Tabel A.1 Data responden (1).....	187
Tabel A.2 Data responden (2).....	188
Tabel A.3 Data responden (3).....	190
Tabel A.4 Data terapi (1).....	192
Tabel A.5 Data terapi (2).....	194
Tabel A.6 Data terapi (3).....	196
Tabel A.7 Data terapi uji coba <i>immersive</i>	198
Tabel A.8 Data kepuasan terapi.....	200

BAB I

PENDAHULUAN

Pada bab ini penulis menjelaskan tentang latar belakang, rumusan masalah, batasan masalah, tujuan, metodologi, dan sistematika penulisan Tugas Akhir.

1.1 Latar Belakang

Fobia serangga (*insectofobia/entomophobia*) adalah salah satu jenis fobia dimana penderitanya memiliki rasa takut berlebih terhadap serangga. Ketakutan berlebihan ini tidak jarang menyebabkan rasa cemas dan kepanikan yang parah. Fobia serangga biasanya timbul akibat adanya rasa jijik terhadap serangga, takut terkena penyakit, maupun trauma dimasa kecil. Fobia serangga dapat menghambat aktifitas sehari-hari penderitanya. Sering kali penderita fobia serangga dijadikan bahan ejekan atau bahkan *bullying* oleh teman-temannya dengan menakut-nakutinya dengan serangga. Hal ini tentu dapat memperparah kondisi penderita.

Untuk menyembuhkan fobia serangga, terapi paling efektif adalah dengan memberanikan diri untuk menghadapi serangga secara langsung. Namun pada praktiknya, tidak semua penderita mampu mengatasi rasa takutnya saat berhadapan langsung dengan serangga. Untuk mengatasi masalah tersebut, perlu dibuat aplikasi pendukung terapi fobia serangga dimana penderita bisa melihat dan berinteraksi dengan serangga virtual.

Aplikasi pendukung terapi fobia serangga ini akan dibangun menggunakan teknologi *Augmented Reality* dengan menggunakan kamera untuk menempatkan serangga virtual di dunia nyata. Dalam aplikasi ini kita dapat melihat dan berinteraksi dengan serangga virtual dengan cara memukulnya. Dengan dibuatnya aplikasi pendukung terapi fobia serangga diharapkan dapat membantu menyembuhkan penderita fobia serangga.

1.2 Rumusan Masalah

Rumusan masalah yang diangkat dalam dalam Tugas Akhir ini adalah sebagai berikut:

1. Bagaimana menampilkan serangga virtual secara acak?
2. Bagaimana cara menggerakkan serangga virtual?
3. Bagaimana cara memperbesar dan memperkecil ukuran serangga virtual?
4. Bagaimana membuat tangan pengguna dapat berinteraksi dengan objek virtual dengan cara memukul?
5. Bagaimana cara menyimpan dan menampilkan skenario terapi?
6. Bagaimana uji coba untuk mengetahui keefektifan dan kinerja dari aplikasi?

1.3 Batasan Masalah

Berikut beberapa hal yang menjadi batasan masalah dalam Tugas Akhir ini:

1. *Augmented Reality* pada aplikasi ini menggunakan *marker* yang sudah ditentukan untuk menampilkan objek virtual di dunia nyata
2. Aplikasi ini harus dijalankan di atas permukaan yang datar dengan pencahayaan yang cukup
3. Serangga virtual hanya dapat dibunuh diatas *marker*.
4. Aplikasi hanya mampu menyimpan maksimal 10 skenario terapi

1.4 Tujuan

Tujuan dari pembuatan tugas akhir ini antara lain:

1. Membuat aplikasi pendukung terapi fobia serangga yang terlihat realistis
2. Membuat aplikasi pendukung terapi fobia serangga dimana pengguna bisa melihat dan berinteraksi dengan memukul serangga virtual

1.5 Manfaat

Manfaat dari hasil pembuatan tugas akhir ini antara lain:

1. Membantu proses terapi fobia serangga yang sesungguhnya dengan menggunakan serangga virtual sebagai media simulasi
2. Membantu menyembuhkan penderita fobia serangga

1.6 Metodologi

Berikut metodologi yang digunakan dalam Tugas Akhir ini:

A. Penyusunan proposal Tugas Akhir

Proposal tugas akhir ini berisi tentang deskripsi pendahuluan dari tugas akhir yang akan dibuat. Pendahuluan ini terdiri atas hal yang menjadi latar belakang diajukannya usulan tugas akhir, rumusan masalah yang diangkat, batasan masalah untuk tugas akhir, tujuan dari pembuatan tugas akhir, dan manfaat dari hasil pembuatan tugas akhir. Selain itu dijabarkan pula tinjauan pustaka yang digunakan sebagai referensi pendukung pembuatan tugas akhir. Sub bab metodologi berisi penjelasan mengenai tahapan penyusunan tugas akhir mulai dari penyusunan proposal hingga penyusunan buku tugas akhir. Terdapat pula sub bab jadwal kegiatan yang menjelaskan jadwal pengerjaan tugas akhir.

B. Studi Literatur

Tahap studi literatur merupakan tahap pembelajaran dan pengumpulan informasi yang digunakan untuk mengimplementasikan tugas akhir. Tahap ini diawali dengan pengumpulan literatur, diskusi, eksplorasi teknologi, dan pustaka, serta pemahaman dasar teori yang digunakan pada topik tugas akhir. Literatur-literatur yang dimaksud disebutkan sebagai berikut:

1. Terapi Fobia Serangga.
2. *Augmented Reality*.
3. *Game Engine Unity*

4. Vuforia SDK.

C. Perancangan perangkat lunak

Pada tahap ini diawali dengan melakukan analisis awal terhadap permasalahan utama yang muncul pada topik tugas akhir. Kemudian dilakukan perancangan perangkat lunak yang meliputi penentuan data yang akan digunakan dan proses-proses yang akan dilaksanakan. Langkah yang akan digunakan pada tahap ini adalah perancangan simulasi terapi fobia serangga sesuai prosedur terapi.

D. Implementasi

Pada tahap ini dilakukan implementasi untuk pembangunan aplikasi pendukung terapi fobia serangga. Aplikasi ini akan dibangun menggunakan *game engine* Unity dengan C# sebagai bahasa pemrograman untuk *scripting*. Versi SDK yang digunakan adalah 25, yaitu Android Lollipop.

E. Uji coba dan evaluasi

Pada tahap ini dilakukan uji coba dengan menggunakan beberapa macam kondisi untuk mencoba aplikasi bisa berjalan atau tidak. Uji fungsionalitas untuk mengetahui apakah aplikasi sudah memenuhi semua kebutuhan fungsional.

F. Penyusunan buku Tugas Akhir

Pada tahapan ini, penulis menyusun laporan tugas akhir sebagai dokumentasi dari pelaksanaan tugas akhir, yang mencakup seluruh konsep, teori, implementasi, serta hasil yang telah dikerjakan.

1.7 Sistematika Penulisan

Berikut sistematika penulisan buku Tugas Akhir ini:

1. BAB I: PENDAHULUAN

Bab ini berisi latar belakang, rumusan masalah, batasan masalah, tujuan, metodologi, dan sistematika penulisan Tugas Akhir.

2. **BAB II: TINJAUAN PUSTAKA**

Bab ini berisi daftar teori mengenai permasalahan dan algoritma yang digunakan dalam Tugas Akhir ini.

3. **BAB III: ANALISIS DAN PERANCANGAN**

Bab ini membahas analisis dari sistem yang dibuat meliputi analisis permasalahan, deskripsi umum perangkat lunak, spesifikasi kebutuhan, dan identifikasi pengguna. Kemudian membahas rancangan dari sistem yang dibuat meliputi rancangan skenario kasus penggunaan, arsitektur, data, dan antarmuka.

4. **BAB IV: IMPLEMENTASI**

Bab ini membahas implementasi dari rancangan sistem yang dilakukan pada tahap perancangan.

5. **BAB V: UJI COBA DAN EVALUASI**

Bab ini membahas pengujian dari aplikasi yang dibuat dengan melihat keluaran yang dihasilkan oleh aplikasi dan evaluasi ke pengguna untuk mengetahui kemampuan aplikasi. Pengguna dalam tugas ini adalah penderita fobia serangga.

6. **BAB VI: KESIMPULAN DAN SARAN**

Bab ini berisi kesimpulan dari hasil uji coba yang telah dilakukan dan saran mengenai hal-hal yang masih bisa dikembangkan.

(Halaman ini sengaja dikosongkan)

BAB II

TINJAUAN PUSTAKA

Pada bab ini penulis menjelaskan tentang beberapa tinjauan pustaka mengenai permasalahan dan algoritma yang digunakan dalam Tugas Akhir.

2.1 Fobia Serangga

Fobia serangga (*insectofobia/entomophobia*) adalah salah satu jenis fobia dimana penderitanya memiliki rasa takut berlebihan terhadap serangga seperti kecoa, jangkrik, laba-laba, kalajengking, tarantula, ngengat, dan serangga lainnya. Ketakutan berlebihan ini tidak jarang menyebabkan rasa cemas dan kepanikan yang parah. Fobia serangga merupakan salah satu jenis fobia spesifik. Fobia serangga biasanya timbul akibat adanya rasa jijik terhadap serangga, takut terkena penyakit, maupun trauma dimasa kecil. Biasanya penderita fobia serangga memiliki ketakutan spesifik akan satu serangga, seperti takut kecoa atau takut laba-laba.

Ketakutan penderita fobia serangga sering kali tidak sebanding dengan bahaya yang dapat dimunculkan oleh serangga yang ditakutinya. Sebagai hasilnya penderita fobia serangga akan selalu berusaha untuk menghindari bertemu dengan serangga yang ditakutinya. Sebagai contoh orang yang takut dengan ngengat akan menghindari tempat dimana biasanya sering terdapat ngengat, seperti di dapur maupun lemari pakaian. Disaat berada pada tempat tersebut, penderita biasanya akan waspada akan adanya ngengat dan tidak jarang mengira binatang terbang lainnya adalah ngengat. Disaat melihat ngengat, penderita akan merasakan rasa takut yang intens yang dapat meningkat hingga serangan panik. Sebagai hasilnya, penderita akan pergi dari situasi secepatnya. [1]

2.2 Terapi Fobia Serangga

Untuk menghilangkan fobia serangga perlu dilakukan terapi fobia serangga. Terapi fobia serangga pada umumnya dilakukan oleh seorang psikolog maupun lembaga konseling. Untuk menyembuhkan fobia serangga perlu dilakukan terapi dengan cara sebagai berikut:

1. *Hypnotherapy*
Proses penyembuhan fobia serangga dengan cara memberikan sugesti-sugesti dari *hypnotherapist* untuk menghilangkan ketakutannya terhadap serangga
2. Desentiasi sistematis
Penyembuhan fobia serangga dengan cara membuat penderita rileks sehingga membayangkan suasana yang indah dan nyaman, setelah itu didatangkan serangga yang ditakutinya.
3. Abreaksi
Penyembuhan fobia serangga melalui cara pendekatan, mulai dengan foto atau gambar serangga yang ditakutinya, kemudian dilihatkan serangga tersebut dari kejauhan lalu didekatkan secara perlahan
4. *Flooding*
Penyembuhan fobia serangga dengan cara menempatkan penderita dengan serangga yang ditakutinya, sampai ia tidak merasa ketakutan terhadap serangga tersebut.
5. *Reframing*
Proses penyembuhan fobia serangga dengan cara membuat penderita membayangkan kejadian masa lalu awalnya dia mengalami fobia serangga, dilanjutkan dengan menyuruhnya bercerita kepada terapis
6. Obat-obatan
Penyembuhan fobia dengan menggunakan obat-obat kimia. Biasanya tidak dianjurkan untuk penderita fobia. Penggunaan obat digunakan sebagai pendamping ketika rasa cemas dan takut datang. Obat yang digunakan biasanya adalah antidepresan dan *beta-blocker*. [2]

2.3 *Augmented Reality*

Augmented Reality dideskripsikan sebagai kombinasi teknologi yang memungkinkan penggabungan antara gambar komputer dengan tampilan video dunia nyata. AR dibangun berbasis teknologi yang digunakan pada *Virtual Reality* (VR) dan dapat berinteraksi tidak hanya dengan dunia virtual melainkan memiliki hubungan langsung dengan dunia nyata. AR memiliki karakteristik mengkombinasikan dunia virtual dengan dunia nyata, interaksi secara *real time*, dan ditampilkan dalam 3D. Saat ini AR sudah digunakan di berbagai bidang seperti medis, militer, manufaktur, visualisasi, hiburan, robotic, edukasi, dan lain lain. [3]

2.4 SDK

Sebuah Software Development Kit (SDK atau devkit) tipikal merupakan satu set perkakas pengembangan software yang digunakan untuk mengembangkan atau membuat aplikasi untuk paket software tertentu, software framework, hardware platform, sistem komputer, konsol video game, sistem operasi atau platform sejenis lainnya. Ia mencakup mulai dari pemrograman sederhana seperti sebuah application programming interface (API), sampai dengan pemrograman yang lebih rumit dengan hardware yang canggih atau pada sistem embedded termasuk perangkat mobile. [4]

2.5 Unity

Unity merupakan game engine yang dikembangkan oleh Unity Technologies. *Software* ini pertama kali diluncurkan pada tahun 2005. Unity merupakan alat bantu pengembang game dengan kemampuan rendering yang terintegrasi di dalamnya. Dengan menggunakan kecanggihan fitur fiturnya dan juga kecepatan kerja yang tinggi, unity dapat menciptakan program interaktif tidak hanya dalam 2 dimensi, tetapi juga dalam bentuk 3 dimensi. Unity sering disebut sebagai game engine yang multiplatform karena bisa digunakan untuk membuat game di berbagai macam platform

seperti Android, iOS, PC, MAC, Linux, Xbox 360, PS3 dan juga Wii. Unity memiliki sebuah *asset store* merupakan sebuah situs yang memungkinkan kita untuk mendapatkan aset-aset yang dibutuhkan dalam membuat game sendiri. [5]. Dalam mengembangkan game, unity mendukung *scripting* dengan bahasa C# dan Javascript. [6]

2.6 Vuforia

Vuforia ini sendiri merupakan SDK yang dikembangkan oleh Qualcomm untuk membantu pengembang dalam menciptakan aplikasi atau game yang memiliki teknologi Augmented Reality.

SDK ini sendiri memiliki berbagai fitur menarik seperti memindai objek, memindai teks, mengenali bingkai penanda, *virtual button*, mengidentifikasi permukaan objek secara pintar, memindai dengan berbasis awan, mengenali target gambar, mengenali target benda silinder, dan mengenali objek target yang telah ditetapkan.

Vuforia SDK sendiri telah mendukung pengembangan aplikasi untuk perangkat yang berbasis iOS dan Android. Selain itu, Unity Game Engine telah didukung Vuforia SDK sehingga lebih mempermudah untuk mengembangkan game Augmented Reality langsung dengan game engine ini.

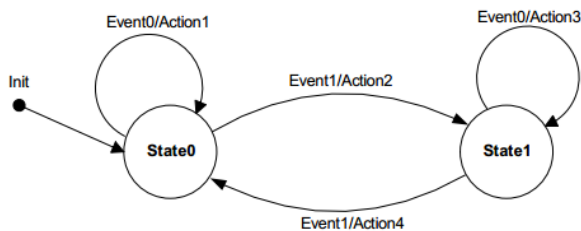
Kini lebih dari 15.000 aplikasi di dunia telah menggunakan Vuforia SDK untuk mengintegrasikan kemampuan fitur Augmented Reality. Selain itu, 150.000 pengembang aplikasi atau game telah terdaftar menggunakan Vuforia SDK ini. [7]

2.7 Finite-State Machine

Finite-State Machine (FSM) adalah sebuah metodologi perancangan sistem kontrol yang menggambarkan tingkah laku atau prinsip kerja sistem dengan menggunakan tiga hal berikut: *State* (Keadaan), *Event* (kejadian) dan *action* (aksi). Pada satu saat dalam periode waktu yang cukup signifikan, sistem akan berada pada salah satu *state* yang aktif. Sistem dapat beralih atau bertransisi menuju *state* lain jika mendapatkan masukan atau *event*

tertentu, baik yang berasal dari perangkat luar atau komponen dalam sistemnya itu sendiri. Transisi keadaan ini umumnya juga disertai oleh aksi yang dilakukan oleh sistem ketika menanggapi masukan yang terjadi. Aksi yang dilakukan tersebut dapat berupa aksi yang sederhana atau melibatkan rangkaian proses yang relatif kompleks

Diagram keadaan pada dasarnya merupakan salah satu bentuk representasi dari FSM. Diagram ini secara visual menggambarkan tingkah laku yang dimiliki oleh sistem kontrol yang kompleks kedalam bentuk yang lebih sederhana dan relative mudah dipahami. Gambar 2.1 menunjukkan contoh diagram state sederhana. [8]



Gambar 2.1 Contoh diagram *state* sederhana

(Halaman ini sengaja dikosongkan)

BAB III

ANALISIS DAN PERANCANGAN

Bab ini menjelaskan tentang analisis dan perancangan aplikasi pendukung Terapi Fobia Serangga menggunakan teknologi *Augmented Reality*. Pembahasan yang akan dilakukan meliputi analisis fitur yang dibutuhkan dan perancangan perangkat lunak.

3.1 Analisis Sistem

Terapi fobia serangga pada umumnya dilakukan oleh seorang psikolog. Proses terapi dimulai dengan mempelajari latar belakang pasien, dilanjutkan dengan pemberian sugesti alam bawah sadar, dan dilanjutkan dengan proses terapi secara bertahap yang dimulai dari pemaparan gambar serangga hingga pemaparan serangga dalam jarak dekat secara bertahap. Namun dalam praktiknya, terapi secara konvensional kurang berhasil karena tidak semua pasien mampu mengatasi rasa takutnya saat berhadapan dengan serangga secara langsung. Maka dari itu dibutuhkan sebuah sistem yang dapat digunakan sebagai media simulasi terapi fobia serangga dengan menggunakan serangga virtual, sehingga pasien lebih dapat mempersiapkan mental saat harus menghadapi serangga yang sebenarnya.

Aplikasi ini dibangun dengan tujuan untuk mendukung proses terapi fobia serangga yang sebenarnya dengan menampilkan serangga virtual dan simulasi membunuh serangga yang dipaparkan dalam bentuk skenario yang dapat ditambahkan sesuai dengan kebutuhan terapi. Serangga virtual yang digunakan didapatkan dari Unity Asset Store, sedangkan pembuatan aplikasi dibuat menggunakan aplikasi Unity. Aplikasi ini berupa aplikasi *Augmented Reality* dimana membutuhkan *marker* untuk menampilkan objek virtual di dunia nyata.

3.2 Perancangan Model 3D

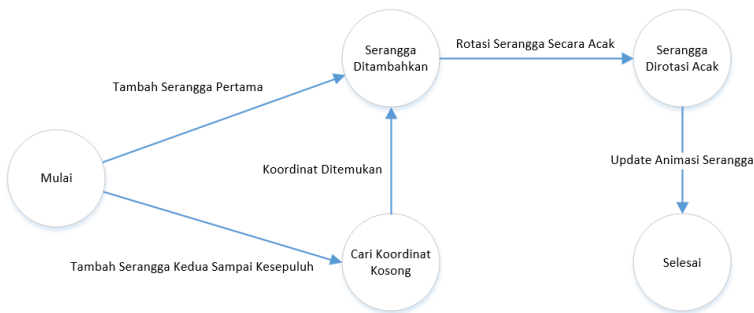
Model yang digunakan dalam aplikasi ini didapatkan dari Unity Asset Store. Model yang didapatkan sudah memiliki tekstur dan animasi idle, berjalan, dan mati. Model yang didapatkan dilakukan beberapa penyesuaian agar dapat ditampilkan dengan baik di aplikasi.

3.3 Perancangan Perilaku Serangga

Pada subbab ini akan dijelaskan rancangan perilaku serangga yang digunakan pada aplikasi ini. Rancangan perilaku serangga akan ditampilkan dalam bentuk FSM.

3.3.1 Rancangan Perilaku Menambah Serangga

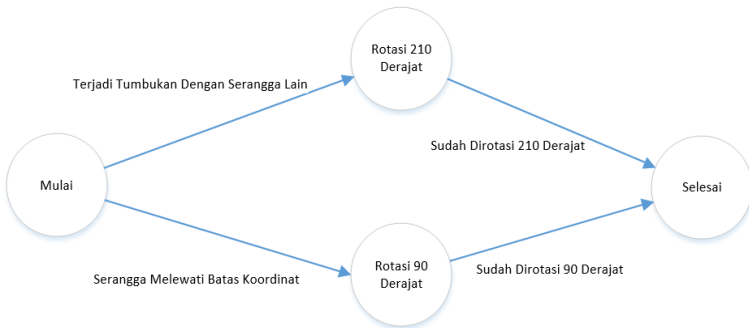
Berikut akan dijelaskan rancangan perilaku menambah serangga dalam bentuk FSM. Penjelasan dapat dilihat pada Gambar 3.1.



Gambar 3.1 FSM menambah serangga

3.3.2 Rancangan Perilaku Serangga Berjalan

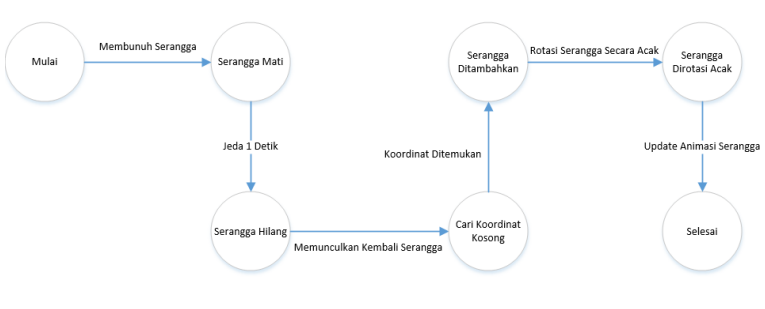
Berikut akan dijelaskan rancangan perilaku serangga berjalan dalam bentuk FSM. Penjelasan dapat dilihat pada Gambar 3.2.



Gambar 3.2 FSM srangga berjalan

3.3.3 Rancangan Perilaku Membunuh Serangga

Berikut akan dijelaskan rancangan perilaku membunuh serangga dalam bentuk FSM. Penjelasan dapat dilihat pada Gambar bla dan Gambar 3.3.



Gambar 3.3 FSM membunuh serangga

3.4 Perancangan Algoritma

Pada subbab ini akan dijelaskan rancangan algoritma yang akan digunakan untuk memenuhi kebutuhan fungsionalitas perangkat lunak yang dibangun. Berikut ini adalah penjelasan dari rancangan algoritma yang digunakan dalam perangkat lunak ini.

3.4.1 Rancangan Algoritma Penambahan Serangga

Pada saat menambahkan serangga, dibutuhkan algoritma agar serangga yang ditambahkan dapat ditempatkan pada lokasi yang tidak saling bertumpukkan. Algoritma yang digunakan dibagi menjadi 2, yaitu algoritma penambahan serangga pertama dan algoritma penambahan serangga kedua hingga kesepuluh. Pada algoritma penambahan serangga pertama dilakukan dengan mengaktifkan objek serangga pertama rotasi acak. Pada algoritma penambahan serangga kedua hingga kesepuluh dilakukan pencarian lokasi yang tidak terdapat *collider* yang terdapat pada objek serangga yang aktif. Setelah didapatkan lokasi yang tepat, dilanjutkan dengan mengaktifkan objek serangga dengan rotasi acak. Algoritma penambahan serangga ini secara lengkap dapat dilihat pada Gambar 3.4 dan Gambar 3.5.

1. Aktifkan objek serangga pertama
2. Aktifkan salah satu child objek serangga pertama sesuai index jenis serangga yang dipilih
3. Set rotasi objek serangga pertama secara acak
4. Perbarui jumlah serangga +1
5. Set animasi berdasarkan pergerakan serangga terakhir

Gambar 3.4 Pseudocode algoritma penambahan serangga pertama

1. Cari lokasi yang tidak terdapat serangga (*collider*)
2. Jika sudah dapat lokasi yang tepat, aktifkan objek serangga ke-n
3. Aktifkan salah satu child objek serangga ke-n sesuai indeks jenis serangga yang dipilih
4. Set rotasi objek serangga ke-n secara acak
5. Perbarui jumlah serangga +1
6. Set animasi berdasarkan pergerakan serangga terakhir

Gambar 3.5 Pseudocode algoritma penambahan serangga kedua hingga kesepuluh

3.4.2 Rancangan Algoritma Pengurangan Serangga

Pengurangan serangga dilakukan dengan menonaktifkan objek serangga yang paling akhir. Algoritma pengurangan serangga ini secara lengkap dapat dilihat pada Gambar 3.6.

1. Cek apakah ada serangga yang ditampilkan
2. Jika ada, nonaktifkan objek serangga yang paling akhir
3. Perbarui jumlah serangga -1

Gambar 3.6 Pseudocode algoritma pengurangan serangga

3.3.4 Algoritma Menjalankan Serangga

Algoritma menjalankan serangga dibutuhkan agar serangga dapat dijalankan dengan baik. Algoritma yang digunakan adalah mengaktifkan *script* untuk menggerakkan serangga dan memperbarui animasi serangga yang ditampilkan menjadi animasi berjalan. Algoritma menjalankan serangga ini secara lengkap dapat dilihat pada Gambar 3.7.

1. Aktifkan *script* untuk menggerakkan serangga
2. Ubah animasi serangga yang ditampilkan menjadi berjalan
3. Perbarui status pergerakan serangga pada menu pengaturan menjadi serangga berjalan

Gambar 3.7 Pseudocode algoritma menjalankan serangga

3.3.5 Algoritma Menghentikan Serangga

Algoritma menghentikan serangga dibutuhkan agar dapat menghentikan serangga dengan baik. Algoritma yang digunakan untuk menghentikan serangga adalah dengan menonaktifkan script untuk menggerakkan serangga dan memperbarui animasi serangga yang ditampilkan menjadi diam. Algoritma menghentikan serangga ini secara lengkap dapat dilihat pada Gambar 3.8.

1. Nonaktifkan *script* untuk menggerakkan serangga
2. Ubah animasi serangga yang ditampilkan menjadi diam
3. Perbarui status pergerakan serangga pada menu pengaturan menjadi serangga diam

Gambar 3.8 Pseudocode algoritma menghentikan serangga

3.3.6 Algoritma Pertumbukkan Serangga

Algoritma pertumbukkan serangga bertujuan agar serangga yang saling bertumbukan tidak terus berjalan menembus satu sama lain, melainkan bergerak ke arah lain. Algoritma yang digunakan adalah dengan mengaktifkan trigger pada *collider* saat terjadi tumbukan dengan serangga lain yang terdapat *rigidbody*. Selanjutnya serangga yang bertumbukan akan di rotasi 210 derajat agar dapat berjalan ke arah lain. Algoritma pertumbukan serangga ini secara lengkap dapat dilihat pada Gambar 3.9.

1. Cek apakah trigger pada serangga yang ditampilkan aktif atau tidak
2. Jika aktif, rotasi serangga yang triggernya aktif tersebut 210 derajat

Gambar 3.9 Pseudocode algoritma pertumbukkan serangga

3.3.7 Algoritma Pembatasan Pergerakan Serangga

Algoritma pembatasan pergerakan serangga bertujuan agar serangga tersebut tidak berjalan terus ke satu arah sampai pada akhirnya menghilang dari *frame* kamera karena berjalan terlalu jauh. Algoritma yang digunakan adalah dengan membatasi koordinat yang dapat dilalui serangga. Jika serangga berjalan melewati batas koordinat tersebut akan dirotasi 90 derajat sehingga tidak akan keluar dari batas koordinat yang sudah ditentukan. Algoritma pembatasan pergerakan serangga ini secara lengkap dapat dilihat pada Gambar 3.10.

1. Cek apakah serangga berjalan melewati batas koordinat atau tidak
2. Jika melewati batas koordinat, serangga tersebut akan dirotasi 90 derajat

Gambar 3.10 Pseudocode algoritma pembatasan pergerakan serangga

3.3.8 Algoritma Pembesaran Serangga

Algoritma pembesaran serangga diperlukan agar serangga dapat diperbesar dengan baik. Algoritma yang digunakan adalah dengan memperbesar dengan skala yang sudah ditentukan dan disertai dengan ukuran maksimal serangga yang diperbolehkan. Skala perbesaran yang digunakan adalah +0.03 dalam satu kali perbesaran. Perbesaran dapat dilakukan hingga 5 kali perbesaran. Algoritma perbesaran serangga ini secara lengkap dapat dilihat pada Gambar 3.11.

1. Cek apakah ukuran serangga sudah maksimal atau belum
2. Jika ukuran serangga belum maksimal, perbesar serangga dengan menambahkan 0.03 dari ukuran serangga sebelumnya (1 kali perbesaran)
3. Perbarui skala ukuran serangga di menu pengaturan +1

Gambar 3.11 Pseudocode algoritma perbesaran serangga

3.3.9 Algoritma Pengecilan Serangga

Algoritma pengecilan serangga diperlukan agar serangga dapat diperkecil dengan baik. Algoritma yang digunakan adalah dengan memperkecil dengan skala yang sudah ditentukan dan disertai dengan ukuran minimal serangga yang diperbolehkan. Skala perkecilan yang digunakan adalah -0.03 dalam satu kali pengecilan. Pengecilan dapat dilakukan hingga 4 kali pengecilan dari skala maksimal yaitu 5. Algoritma pengecilan serangga ini secara lengkap dapat dilihat pada Gambar 3.12.

1. Cek apakah ukuran serangga sudah minimal atau belum
2. Jika ukuran serangga belum minimal, perkecil serangga dengan mengurangi 0.03 dari ukuran serangga sebelumnya (1 kali pengecilan)
3. Perbarui skala ukuran serangga di menu pengaturan -1

Gambar 3.12 Pseudocode algoritma pengecilan serangga

3.3.10 Rancangan Algoritma Bunuh Serangga

Pada saat membunuh serangga diperlukan algoritma agar serangga dapat dibunuh dengan cara dipukul dan ditampilkan dengan baik. Algoritma yang digunakan adalah dengan menambahkan virtual button pada setiap serangga dan mengecek virtual button mana yang tersentuh saat kita membunuh serangga dengan cara memukulnya. Saat tangan kita masih berada di atas serangga yang kita bunuh, animasi serangga yang kita bunuh diperbarui menjadi mati dan dilanjutkan dengan menonaktifkan *script* untuk menjalankan serangga. Setelah tangan kita sudah kita lepas dari serangga yang kita bunuh, dilanjutkan dengan mengaktifkan *coroutine* selama 1 detik dan dilanjutkan dengan proses pemunculan lagi serangga yang kita bunuh. Pada proses tersebut, dilakukan pengecekan serangga mana yang mati, kemudian dilakukan pencarian lokasi yang tidak terdapat *collider* yang terdapat pada serangga yang aktif. Setelah itu serangga ditampilkan di lokasi barunya dengan rotasi acak dengan animasi

sesuai dengan pergerakan terakhir. Algoritma bunuh serangga ini secara lengkap dapat dilihat pada Gambar 3.13.

1. Cek virtual button pada serangga mana yang tertekan (tangan berada diatas serangga)
2. Ubah animasi serangga yang terbunuh jadi mati
3. Matikan script untuk menjalankan serangga
4. Cek jika virtual button sudah tidak tertekan (tangan lepas dari serangga)
5. Aktifkan korutin 1 detik
6. Perbarui status serangga sudah mati
7. Jalankan fungsi pemunculan lagi serangga yang kita bunuh
8. Cek serangga mana yang mati
9. Cari lokasi pemunculan baru yang tidak terdapat *collider* yang terdapat pada serangga yang aktif
10. Menampilkan serangga yang tadi kita bunuh di lokasi barunya dengan rotasi acak
11. Set animasi berdasarkan pergerakan terakhir

Gambar 3.13 Pseudocode algoritma bunuh serangga

3.3.11 Rancangan Algoritma Mengembalikan Posisi Serangga Pertama

Pada saat mengembalikan posisi serangga pertama ke posisi awal yaitu di tengah *marker* diperlukan algoritma agar posisi serangga dapat dikembalikan ke posisi awal dengan baik. Algoritma yang digunakan adalah dengan mendeskripsikan posisi tengah marker yaitu pada koordinat (0,0,0) dan memindahkan serangga pertama ke posisi tersebut. Algoritma mengembalikan posisi serangga pertama ini secara lengkap dapat dilihat pada Gambar 3.14.

1. Deskripsikan koordinat tengah *marker* (0,0,0)
2. Pindahkan posisi serangga pertama ke koordinat tersebut

Gambar 3.14 Pseudocode algoritma mengembalikan posisi serangga pertama

3.3.12 Rancangan Algoritma Penambahan Skenario

Pada saat menambahkan skenario diperlukan algoritma agar skenario dapat ditambahkan dengan baik. Algoritma yang digunakan adalah dengan menyimpan data skenario ke dalam array *scenarioTemp* sebagai penyimpanan sementara. Kemudian jika semua aksi skenario sudah ditambahkan maka data akan disimpan di *playerprefs* dengan menggunakan satu *key* untuk setiap skenario, yaitu menggunakan *key* skenario1, skenario2, dan seterusnya hingga skenario10. Algoritma penambahan skenario ini secara lengkap dapat dilihat pada Gambar 3.15.

1. Pengguna mengisi *form* aksi, *value*, dan durasi pada menu penambahan skenario
2. Mengambil data aksi, *value*, dan durasi dari *form* penambahan skenario
3. Menyimpan data aksi, *value*, dan durasi ke *array scenarioTemp* sebagai penyimpanan sementara
4. Jika aksi ditambahkan data aksi berikutnya akan ditambahkan pada *array scenarioTemp* dengan indeks selanjutnya
5. Setelah semua aksi sudah ditambahkan data seluruh aksi yang terdapat pada *array scenarioTemp* disimpan di *playerprefs* dengan *key* sesuai skenario yang ditambahkan

Gambar 3.15 Pseudocode algoritma penambahan skenario

3.3.13 Rancangan Algoritma Pengubahan Skenario

Pada saat mengubah skenario diperlukan algoritma agar skenario dapat diubah dengan baik. Algoritma yang digunakan adalah dengan mengambil data skenario yang dipilih dari *playerprefs*. Kemudian dilakukan pengubahan data skenario yang dipilih. Selanjutnya data akan disimpan dalam array *scenarioTemp* sebagai penyimpanan sementara. Kemudian jika semua perubahan sudah dilakukan maka data akan disimpan di *playerprefs* dengan menggunakan *key* sesuai dengan skenario yang diubah. Algoritma pengubahan skenario ini secara lengkap dapat dilihat pada Gambar 3.16.

1. Pengguna memilih skenario yang ingin diubah
2. Mengambil data skenario yang dipilih dari *playerprefs* berdasarkan indeks skenario tersebut
3. Pengguna melakukan perubahan data
4. Mengambil data aksi, *value*, dan durasi dari *form* perubahan skenario
5. Menyimpan data aksi, *value*, dan durasi ke *array scenarioTemp* sebagai penyimpanan sementara
6. Jika aksi ditambahkan data aksi berikutnya akan ditambahkan pada *array scenarioTemp* dengan indeks selanjutnya
7. Setelah semua perubahan data sudah dilakukan data seluruh aksi yang terdapat pada *array scenarioTemp* disimpan di *playerprefs* dengan *key* sesuai skenario yang diubah

Gambar 3.16 Pseudocode algoritma perubahan skenario

3.3.14 Rancangan Algoritma Penghapusan Skenario

Pada saat menghapus skenario diperlukan algoritma agar skenario dapat dihapus dengan baik. Algoritma yang digunakan adalah dengan melakukan penghapusan data pada *playerprefs* berdasarkan *key* skenario yang akan dihapus. Algoritma penghapusan skenario ini secara lengkap dapat dilihat pada Gambar 3.17.

1. Pengguna memilih skenario yang ingin dihapus
2. Melakukan penghapusan data pada *playerprefs* berdasarkan *key* skenario yang akan dihapus

Gambar 3.17 Pseudocode algoritma penambahan skenario

3.3.15 Rancangan Algoritma Menjalankan Skenario

Pada saat menjalankan skenario diperlukan algoritma agar skenario dapat dijalankan dengan baik. Algoritma yang digunakan adalah dengan mengambil data skenario yang dipilih yang tersimpan pada *playerprefs* dengan *key* skenario yang dipilih dan

menjalankannya dalam sebuah *corroutine*. Kemudian dilakukan pengecekan aksi, *value*, dan durasi pada skenario yang dipilih. Selanjutnya dijalankan aksi berdasarkan *value* dan durasi yang tersimpan pada setiap aksi yang dijalankan. Algoritma menjalankan skenario ini secara lengkap dapat dilihat pada Gambar 3.18.

- | |
|---|
| <ol style="list-style-type: none"> 1. Pengguna memilih skenario yang ingin dijalankan 2. Mengambil data skenario yang tersimpan pada <i>playerprefs</i> dengan <i>key</i> skenario yang dipilih. 3. Skenario dijalankan pada sebuah <i>corroutine</i> 4. Mengecek jumlah aksi, <i>value</i>, dan, durasi pada skenario yang dipilih 5. Menjalankan aksi berdasarkan <i>value</i> dan durasi yang tersimpan |
|---|

Gambar 3.18 Pseudocode algoritma menjalankan skenario

3.3.16 Rancangan Algoritma Menghentikan Skenario

Pada saat menghentikan skenario diperlukan algoritma agar skenario dapat dihentikan dengan baik. Algoritma yang digunakan adalah dengan menghentikan semua *corroutine* skenario yang sedang dijalankan dan menghilangkan serangga yang ditampilkan saat skenario dijalankan. Algoritma menghentikan skenario ini secara lengkap dapat dilihat pada Gambar 3.19.

- | |
|--|
| <ol style="list-style-type: none"> 1. Menghentikan semua <i>corroutine</i> yang sedang dijalankan 2. Menghilangkan serangga yang ditampilkan |
|--|

Gambar 3.19 Pseudocode algoritma menjalankan skenario

3.5 Perancangan Perangkat Lunak

Pada subbab ini akan dibahas mengenai deskripsi umum perangkat lunak, spesifikasi kebutuhan fungsional dan kebutuhan non-fungsional serta bagaimana karakteristik pengguna aplikasi.

3.5.1 Deskripsi Umum Perangkat Lunak

Dalam tugas akhir ini, dibuat aplikasi pendukung terapi fobia serangga menggunakan teknologi *Augmented Reality*. Aplikasi ini dibangun berbasis Android. Untuk menjalankan aplikasi ini dibutuhkan *marker* sebagai penanda untuk menampilkan serangga virtual di dunia nyata. Gambaran diagram arsitektur sistem dapat dilihat pada Gambar 3.20.



Gambar 3.20 Diagram arsitektur sistem

Secara umum aplikasi ini akan menampilkan serangga virtual. Terdapat 5 jenis serangga yang bisa ditampilkan, yaitu laba-laba, kecoa, kalajengking, tarantula, dan jangkrik. Serangga virtual dapat ditampilkan dalam keadaan diam dan berjalan. Ukuran serangga virtual dapat diperbesar dan diperkecil. Serangga virtual dapat dibunuh dengan cara dipukul. Aplikasi ini juga dapat menyimpan skenario terapi dengan jumlah maksimal 10 skenario. Skenario ditambahkan berdasarkan aksi yang dipilih, nilai pada aksi yang dipilih, dan durasi berlangsungnya aksi. Skenario dapat ditambahkan dengan menampilkan beberapa aksi. Skenario dapat

ditambahkan, diubah, dan dihapus. Skenario yang tersimpan dapat dijalankan dan dihentikan.

3.5.2 Spesifikasi Kebutuhan Fungsional

Berdasarkan deskripsi umum sistem, maka dapat disimpulkan bahwa kebutuhan fungsional dari aplikasi ini ada 14 buah yaitu dapat menambah serangan, mengurangi serangan, menjalankan serangan, menghentikan serangan, memperbesar serangan, memperkecil serangan, mengaktifkan fitur bunuh serangan, menonaktifkan fitur bunuh serangan, mengembalikan posisi serangan pertama, menambah skenario, mengubah skenario, menghapus skenario, menjalankan skenario, dan menghentikan skenario.

3.5.3 Spesifikasi Kebutuhan Non-Fungsional

Terdapat beberapa kebutuhan non-fungsional yang apabila dipenuhi, dapat meningkatkan kualitas dari aplikasi ini. Berikut daftar kebutuhan non-fungsional:

1. Kebutuhan Grafis
Keberhasilan aplikasi dalam membantu terapi sangat bergantung pada kualitas model serangan yang ditampilkan. Semakin mirip serangan virtual yang ditampilkan akan memberikan kesan bahwa serangan yang ditampilkan nyata.

3.5.4 Karakteristik Pengguna

Pengguna yang akan menggunakan aplikasi ini berjumlah dua orang, yaitu terapis yang membantu proses terapi serangan dan pasien yang akan menjalani terapi. Karakteristik pengguna tercantum dalam Tabel 3.1

Tabel 3.1 Karakteristik pengguna

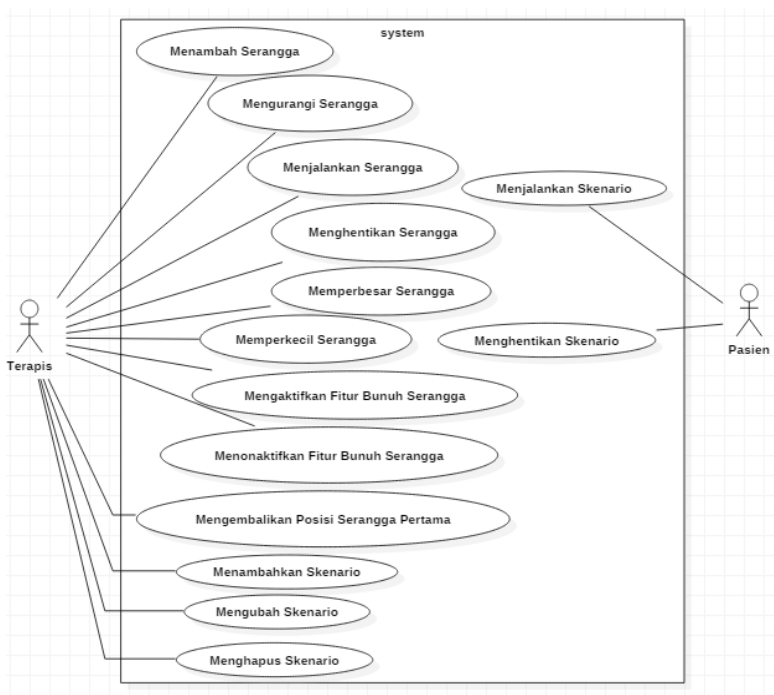
Nama Aktor	Tugas	Hak Akses Aplikasi	Kemampuan yang harus dimiliki
Terapis	Menjalankan aplikasi untuk membantu pasien melakukan terapi fobia serangga. Menambahkan skenario terapi sesuai kebutuhan terapi.	Menjalankan aplikasi	Menjalankan aplikasi sesuai prosedur terapi
Pasien	Menjalankan aplikasi untuk melakukan terapi fobia serangga secara mandiri	Menjalankan aplikasi	Menjalankan aplikasi berdasarkan skenario terapi yang diperlukan

3.6 Perancangan Sistem

Tahap perancangan dalam subbab ini dibagi menjadi beberapa bagian yaitu perancangan diagram kasus penggunaan, perancangan skenario kasus penggunaan, perancangan arsitektur, perancangan antarmuka pengguna, dan perancangan kontrol aplikasi.

3.6.1 Perancangan Diagram Kasus Penggunaan

Dalam aplikasi ini, terdapat 14 kasus penggunaann. Rancangan kasus penggunaan dapat dilihat pada Gambar 3.21.



Gambar 3.21 Diagram kasus penggunaan

Penjelasan singkat dari masing masing kasus penggunaan dapat dilihat pada Tabel 3.2

Tabel 3.2 Skenario kasus penggunaan

No	Kode Kasus Penggunaan	Nama Kasus Penggunaan	Keterangan
1	UC-001	Menambah serangga	Pengguna dapat menambah serangga virtual yang ditampilkan hingga maksimal 10 serangga
2	UC-002	Mengurangi serangga	Pengguna dapat mengurangi serangga virtual yang ditampilkan hingga minimal 0 serangga
3	UC-003	Menjalankan serangga	Pengguna dapat menjalankan serangga virtual yang ditampilkan
4	UC-004	Menghentikan serangga	Pengguna dapat menghentikan serangga virtual yang ditampilkan
5	UC-005	Memperbesar Serangga	Pengguna dapat memperbesar serangga virtual yang ditampilkan hingga 4 kali perbesaran dengan skala dari 1 sampai 5
6.	UC-006	Memperkecil serangga	Pengguna dapat memperkecil serangga virtual yang ditampilkan hingga 4 kali pengecilan dengan skala dari 1 sampai 5

7.	UC-007	Mengaktifkan mode bunuh serangga	Pengguna dapat mengaktifkan mode bunuh serangga agar serangga virtual dapat dibunuh dengan cara dipukul dengan tangan
8.	UC-008	Menonaktifkan mode bunuh serangga	Pengguna dapat menonaktifkan mode bunuh serangga agar serangga tidak dapat dibunuh jika tersentuh tangan
9	UC-009	Mengembalikan Posisi Serangga Pertama	Pengguna dapat mengembalikan posisi serangga pertama ke tengah marker
10	UC-010	Menambah Skenario	Pengguna dapat menambahkan skenario dengan jumlah maksimal 10 skenario
11	UC-011	Mengubah Skenario	Pengguna dapat mengubah skenario yang dipilih
12	UC-012	Menghapus Skenario	Pengguna dapat menghapus skenario yang dipilih
13	UC-013	Menjalankan Skenario	Pengguna dapat menjalankan skenario yang dipilih
14	UC-014	Menghentikan Skenario	Pengguna dapat menghentikan skenario yang dijalankan

3.6.2 Perancangan Skenario Kasus Penggunaan

Setelah diketahui bagaimana perancangan kasus penggunaan, langkah selanjutnya adalah membuat skenario kasus penggunaan beserta diagram aktivitasnya.

Skenario kasus penggunaan dibagi menjadi tiga bagian sesuai dengan kasus penggunaan yang ada dan tercantum pada Tabel 3.3 sampai dengan Tabel 3.16. Tabel tersebut berisi penjelasan skenario yang akan dilakukan ketika pengujian.

Tabel 3.3 Skenario kasus penggunaan menambah serangga

Nama Kasus Penggunaan	Menambah Serangga
Kode	UC-001
Deskripsi	Kasus penggunaan agar aktor dapat menambah serangga virtual yang ditampilkan
Aktor	Terapis
Kondisi Awal	Pengguna sudah masuk ke menu pengaturan pada mode terapi manual
Alur Normal	<ol style="list-style-type: none"> 1. Pengguna menyentuh tombol tambah serangga 2. Sistem menampilkan 5 pilihan jenis serangga yang dapat ditambahkan 3. Pengguna menyentuh tombol laba laba <ol style="list-style-type: none"> A1. Pengguna menyentuh tombol kecoa A2. Pengguna menyentuh tombol kalajengking A3. Pengguna menyentuh tombol tarantula A4. Pengguna menyentuh tombol jangkrik 4. Sistem menampilkan serangga yang ditambahkan pengguna
Alur Alternatif	<ol style="list-style-type: none"> A1. Pengguna menyentuh tombol kecoa <ol style="list-style-type: none"> 1. Kembali ke alur kejadian nomor 4 A2. Pengguna menyentuh tombol kalajengking <ol style="list-style-type: none"> 1. Kembali ke alur kejadian nomor 4 A3. Pengguna menyentuh tombol tarantula <ol style="list-style-type: none"> 1. Kembali ke alur kejadian nomor 4 A4. Pengguna menyentuh tombol jangkrik <ol style="list-style-type: none"> 1. Kembali ke alur kejadian nomor 4

Tabel 3.4 Skenario kasus penggunaan mengurangi serangan

Nama Kasus Penggunaan	Mengurangi Serangga
Kode	UC-002
Deskripsi	Kasus penggunaan agar aktor dapat mengurangi jumlah serangga virtual yang ditampilkan
Aktor	Terapis
Kondisi Awal	Pengguna sudah masuk ke menu pengaturan pada mode terapi manual Sudah ada serangga yang ditampilkan
Alur Normal	<ol style="list-style-type: none"> 1. Pengguna menyentuh tombol kurangi serangga 2. Sistem mengurangi 1 serangga virtual yang ditampilkan
Alur Alternatif	-

Tabel 3.5 Skenario kasus penggunaan menjalankan serangan

Nama Kasus Penggunaan	Menjalankan Serangga
Kode	UC-003
Deskripsi	Kasus penggunaan agar aktor dapat menjalankan serangga virtual yang ditampilkan
Aktor	Terapis
Kondisi Awal	Pengguna sudah masuk ke menu pengaturan pada mode terapi manual Sudah ada serangga yang ditampilkan
Alur Normal	<ol style="list-style-type: none"> 1. Pengguna menyentuh tombol serangga berjalan 2. Sistem menampilkan serangga virtual yang berjalan
Alur Alternatif	-

Tabel 3.6 Skenario kasus penggunaan menghentikan serangga

Nama Kasus Penggunaan	Menghentikan Serangga
Kode	UC-004
Deskripsi	Kasus penggunaan agar aktor dapat menghentikan serangga virtual yang ditampilkan
Aktor	Terapis
Kondisi Awal	Pengguna sudah masuk ke menu pengaturan pada mode terapi manual Sudah ada serangga yang ditampilkan
Alur Normal	<ol style="list-style-type: none"> 1. Pengguna menyentuh tombol serangga diam 2. Sistem menampilkan serangga virtual dalam keadaan diam
Alur Alternatif	-

Tabel 3.7 Skenario kasus penggunaan memperbesar serangan

Nama Kasus Penggunaan	Memperbesar Serangga
Kode	UC-005
Deskripsi	Kasus penggunaan agar aktor dapat memperbesar ukuran serangga virtual yang ditampilkan
Aktor	Terapis
Kondisi Awal	Pengguna sudah masuk ke menu pengaturan pada mode terapi manual Sudah ada serangga yang ditampilkan
Alur Normal	<ol style="list-style-type: none"> 1. Pengguna menyentuh tombol perbesar serangga 2. Sistem memperbesar serangga virtual yang ditampilkan dengan skala +1
Alur Alternatif	-

Tabel 3.8 Skenario kasus penggunaan memperkecil serangga

Nama Kasus Penggunaan	Memperkecil Serangga
Kode	UC-006
Deskripsi	Kasus penggunaan agar aktor dapat memperkecil ukuran serangga virtual yang ditampilkan
Aktor	Terapis
Kondisi Awal	Pengguna sudah masuk ke menu pengaturan pada mode terapi manual Sudah ada serangga yang ditampilkan
Alur Normal	<ol style="list-style-type: none"> 1. Pengguna menyentuh tombol perkecil serangga 2. Sistem memperkecil serangga virtual yang ditampilkan dengan skala -1
Alur Alternatif	-

Tabel 3.9 Skenario kasus penggunaan mengaktifkan fitur bunuh serangga

Nama Kasus Penggunaan	Mengaktifkan Fitur Bunuh Serangga
Kode	UC-007
Deskripsi	Kasus penggunaan agar aktor dapat mengaktifkan fitur bunuh serangga
Aktor	Terapis
Kondisi Awal	Pengguna sudah masuk ke menu pengaturan pada mode terapi manual
Alur Normal	<ol style="list-style-type: none"> 1. Pengguna menyentuh tombol bunuh serangga: on 2. Sistem mengaktifkan fitur bunuh serangga
Alur Alternatif	-

Tabel 3.10 Skenario kasus penggunaan menonaktifkan fitur bunuh serangga

Nama Kasus Penggunaan	Menonaktifkan Fitur Bunuh Serangga
Kode	UC-008
Deskripsi	Kasus penggunaan agar aktor dapat menonaktifkan fitur bunuh serangga
Aktor	Terapis
Kondisi Awal	Pengguna sudah masuk ke menu pengaturan pada mode terapi manual
Alur Normal	<ol style="list-style-type: none"> 1. Pengguna menyentuh tombol bunuh serangga: off 2. Sistem menonaktifkan fitur bunuh serangga
Alur Alternatif	-

Tabel 3.11 Skenario kasus penggunaan mengembalikan posisi serangga pertama

Nama Kasus Penggunaan	Mengembalikan Posisi Serangga Pertama
Kode	UC-009
Deskripsi	Kasus penggunaan agar aktor dapat mengembalikan posisi serangga pertama ke tengah <i>marker</i>
Aktor	Terapis
Kondisi Awal	Pengguna sudah masuk ke menu pengaturan pada mode terapi manual Sudah ada serangga yang ditampilkan
Alur Normal	<ol style="list-style-type: none"> 1. Pengguna menyentuh tombol reset 2. Sistem mengembalikan posisi serangga pertama ke tengah <i>marker</i>
Alur Alternatif	-

Tabel 3.12 Skenario kasus penggunaan menambah skenario

Nama Kasus Penggunaan	Menambah Skenario
Kode	UC-010
Deskripsi	Kasus penggunaan agar aktor dapat menambahkan skenario
Aktor	Terapis
Kondisi Awal	Pengguna sudah masuk ke menu terapiis Jumlah skenario yang tersimpan belum 10
Alur Normal	<ol style="list-style-type: none"> 1. Pengguna menyentuh tombol kelola skenario 2. Sistem menampilkan menu skenario 3. Pengguna menyentuh tombol tambah skenario 4. Sistem menampilkan isian tambah skenario 5. Pengguna mengisi isian menambah skenario 6. Pengguna menyentuh tombol simpan skenario <ol style="list-style-type: none"> A1. Isian durasi belum diisi A2. Isian durasi tidak berisi angka A3. Jumlah serangan total yang akan ditambahkan melebihi 10 A4. Jumlah serangan total yang akan ditambahkan kurang dari 0 A5. Ukuran serangan total yang akan ditambahkan melebihi 5 A6. Ukuran serangan total yang akan ditambahkan kurang dari 1 A7. Pengguna menyentuh tombol tambah aksi selanjutnya 7. Sistem menyimpan skenario

Alur Alternatif	<p>A1. Isian durasi belum diisi</p> <ol style="list-style-type: none"> 1. Sistem menampilkan peringatan data belum lengkap 2. Kembali ke alur kejadian nomor 5 <p>A2. Isian durasi tidak berisi angka</p> <ol style="list-style-type: none"> 1. Sistem menampilkan peringatan isi durasi salah 2. Kembali ke alur kejadian nomor 5 <p>A3. Jumlah serangga total yang akan ditambahkan melebihi 10</p> <ol style="list-style-type: none"> 1. Sistem menampilkan peringatan jumlah serangga terlalu banyak 2. Kembali ke alur kejadian nomor 5 <p>A4. Jumlah serangga total yang akan ditambahkan kurang dari 0</p> <ol style="list-style-type: none"> 1. Sistem menampilkan peringatan jumlah serangga terlalu sedikit 2. Kembali ke alur kejadian nomor 5 <p>A5. Ukuran serangga total yang akan ditambahkan melebihi 5</p> <ol style="list-style-type: none"> 1. Sistem menampilkan peringatan ukuran serangga terlalu besar 2. Kembali ke alur kejadian nomor 5 <p>A6. Ukuran serangga total yang akan ditambahkan kurang dari 1</p> <ol style="list-style-type: none"> 1. Sistem menampilkan peringatan ukuran serangga terlalu kecil 2. Kembali ke alur kejadian nomor 5
------------------------	--

	<p>A7. Pengguna menyentuh tombol tambah aksi serangga</p> <p>1. Kembali ke alur kejadian nomor 4</p>
--	--

Tabel 3.13 Skenario kasus penggunaan mengubah skenario

Nama Kasus Penggunaan	Mengubah Skenario
Kode	UC-011
Deskripsi	Kasus penggunaan agar aktor dapat mengubah skenario yang sudah ditambahkan
Aktor	Terapis
Kondisi Awal	Pengguna sudah masuk ke menu terapi Sudah ada skenario yang ditambahkan
Alur Normal	<ol style="list-style-type: none">1. Pengguna menyentuh tombol kelola skenario2. Sistem menampilkan menu skenario3. Pengguna menyentuh tombol ubah skenario4. Sistem menampilkan menu ubah skenario5. Pengguna menyentuh tombol skenario yang ingin diubah6. Sistem menampilkan isian ubah skenario7. Pengguna melakukan perubahan pada isian skenario8. Pengguna menyentuh tombol simpan skenario<ol style="list-style-type: none">A1. Isian durasi belum diisiA2. Isian durasi tidak berisi angkaA3. Jumlah serangga total yang akan ditambahkan melebihi 10A4. Jumlah serangga total yang ditambahkan kurang dari 0A5. Ukuran serangga total yang akan ditambahkan melebihi 5A6. Ukuran serangga total yang akan ditambahkan kurang dari 1A7. Pengguna menyentuh tombol tambah aksi selanjutnya9. Sistem menyimpan skenario

Alur Alternatif	<p>A1. Isian durasi belum diisi</p> <ol style="list-style-type: none"> 1. Sistem menampilkan peringatan data belum lengkap 2. Kembali ke alur kejadian nomor 7 <p>A2. Isian durasi tidak berisi angka</p> <ol style="list-style-type: none"> 1. Sistem menampilkan peringatan isi durasi salah 2. Kembali ke alur kejadian nomor 7 <p>A3. Jumlah serangga total yang akan ditambahkan melebihi 10</p> <ol style="list-style-type: none"> 1. Sistem menampilkan peringatan jumlah serangga terlalu banyak 2. Kembali ke alur kejadian nomor 7 <p>A4. Jumlah serangga total yang akan ditambahkan kurang dari 0</p> <ol style="list-style-type: none"> 1. Sistem menampilkan peringatan jumlah serangga terlalu sedikit 2. Kembali ke alur kejadian nomor 7 <p>A5. Ukuran serangga total yang akan ditambahkan melebihi 5</p> <ol style="list-style-type: none"> 1. Sistem menampilkan peringatan ukuran serangga terlalu besar 2. Kembali ke alur kejadian nomor 7
------------------------	---

	<p>A6. Ukuran serangga total yang akan ditambahkan kurang dari 1</p> <ol style="list-style-type: none">1. Sistem menampilkan peringatan ukuran serangga terlalu kecil2. Kembali ke alur kejadian nomor 7 <p>A7. Pengguna menyentuh tombol tambah aksi serangga</p> <ol style="list-style-type: none">1. Kembali ke alur kejadian nomor 6
--	---

Tabel 3.14 Skenario kasus penggunaan menghapus skenario

Nama Kasus Penggunaan	Menghapus Skenario
Kode	UC-012
Deskripsi	Kasus penggunaan agar aktor dapat menghapus skenario yang sudah ditambahkan
Aktor	Terapis
Kondisi Awal	Pengguna sudah masuk ke menu terapi Sudah ada skenario yang ditambahkan
Alur Normal	<ol style="list-style-type: none"> 1. Pengguna menyentuh tombol kelola skenario 2. Sistem menampilkan menu skenario 3. Pengguna menyentuh tombol hapus skenario 4. Sistem menampilkan menu hapus skenario 5. Pengguna menyentuh tombol skenario yang ingin dihapus 6. Sistem menghapus skenario
Alur Alternatif	-

Tabel 3.15 Skenario kasus penggunaan menjalankan skenario

Nama Kasus Penggunaan	Menjalankan Skenario
Kode	UC-013
Deskripsi	Kasus penggunaan agar aktor dapat menjalankan skenario yang dipilih
Aktor	Pasien
Kondisi Awal	Pengguna sudah masuk ke menu mulai terapi Sudah ada skenario yang ditambahkan
Alur Normal	<ol style="list-style-type: none"> 1. Pengguna menyentuh tombol mulai terapi 2. Sistem menampilkan menu pilihan skenario 3. Pengguna menyentuh tombol skenario yang ingin dijalankan 4. Sistem menampilkan skenario yang dijalankan
Alur Alternatif	-

Tabel 3.16 Skenario kasus penggunaan menghentikan skenario

Nama Kasus Penggunaan	Menghentikan Skenario
Kode	UC-014
Deskripsi	Kasus penggunaan agar aktor dapat menghentikan skenario yang sedang dijalankan
Aktor	Pasien
Kondisi Awal	Sudah ada skenario yang dijalankan
Alur Normal	<ol style="list-style-type: none"> 1. Pengguna menyentuh stop skenario 2. Sistem menghentikan skenario 3. Sistem menampilkan menu pilihan skenario pasien
Alur Alternatif	-

(Halaman ini sengaja dikosongkan)

BAB IV IMPLEMENTASI

Pada bab ini penulis menjelaskan tentang implementasi berdasarkan desain algoritma serta struktur data yang telah dilakukan.

4.1 Lingkungan Implementasi

Lingkungan implementasi yang digunakan sebagai berikut:

Tabel 4.1 Lingkungan implementasi perangkat lunak

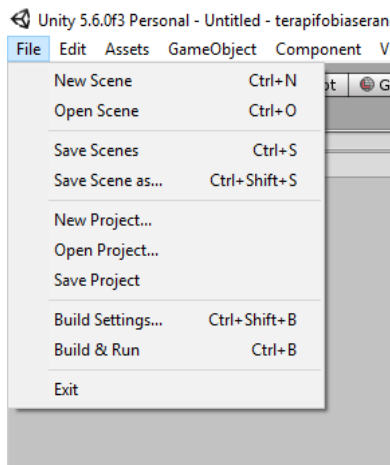
Perangkat Keras	Prosesor : - Intel(R) Core(TM) i5-5200U CPU @ 2.20GHz Memori : - 4 GB - NVIDIA GeForce GT 920M
Perangkat Lunak	Sistem Operasi : - Microsoft Windows 10 Pro 64-bit Perangkat Pengembang : - Unity 5.6.0f3 - Monodevelop 5.9.6

4.2 Implementasi Menambah Serangga

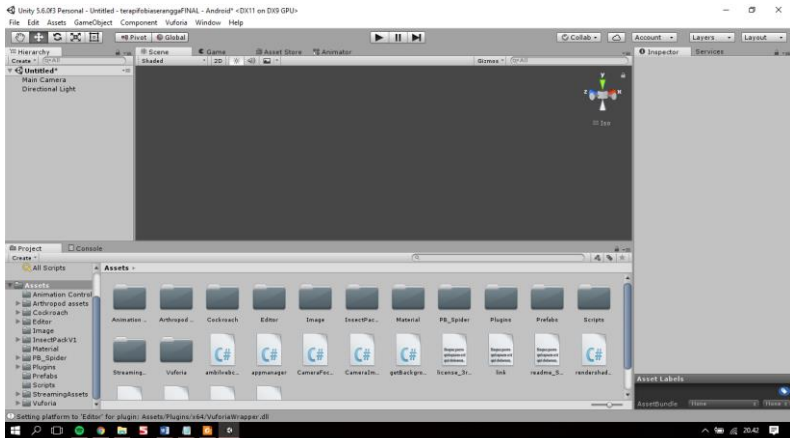
Implementasi menambah serangga dibagi menjadi 5 subbab meliputi pembuatan scene dan load object, script, canvas, dan penambahan serangga.

4.2.1 Implementasi Pembuatan Scene dan Load Objek

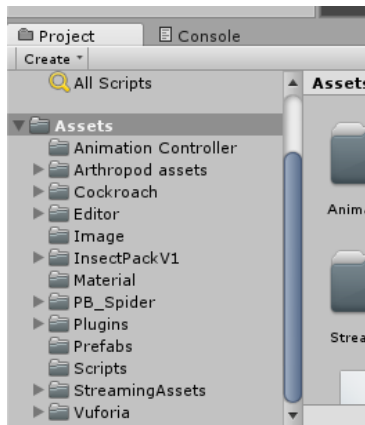
Untuk menambahkan scene, hal yang perlu dilakukan adalah menuju tab “file” lalu pilih “new scene”. Untuk lebih jelasnya, pembuatan scene baru ada di Gambar 4.1. Setelah scene terbuat seperti pada Gambar 4.2, maka objek-objek yang telah kita miliki terdapat pada project explorer yang terletak di kiri bawah aplikasi Unity seperti yang ditunjukkan pada Gambar 4.3. Untuk memasukan objek-objek, berkas yang terdapat dalam proyek bisa dimasukkan dengan cara *drag and drop* berkas dari *map* yang terdapat dalam proyek ke dalam *tab scene* (Penjelasan pada Gambar 4.4). Untuk melihat properti objek mengenai posisi atau yang lainnya, pilih objek yang akan dilihat propertinya. Informasi properti terdapat pada label “*Inspector*”. Lebih jelasnya, ada pada Gambar 4.5.



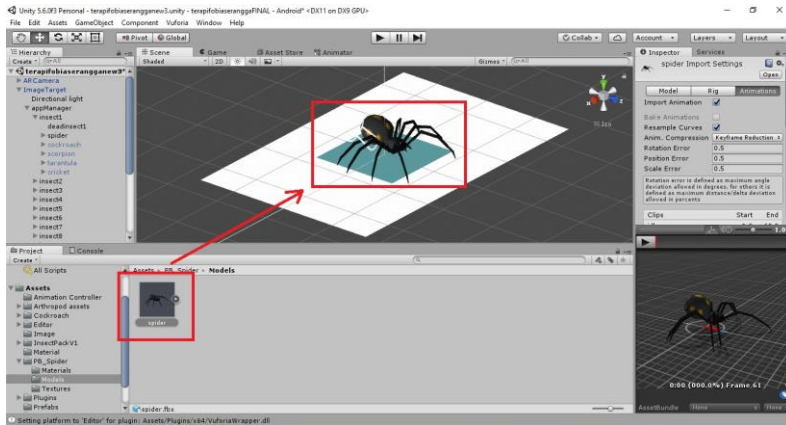
Gambar 4.1 Penambahan *scene*



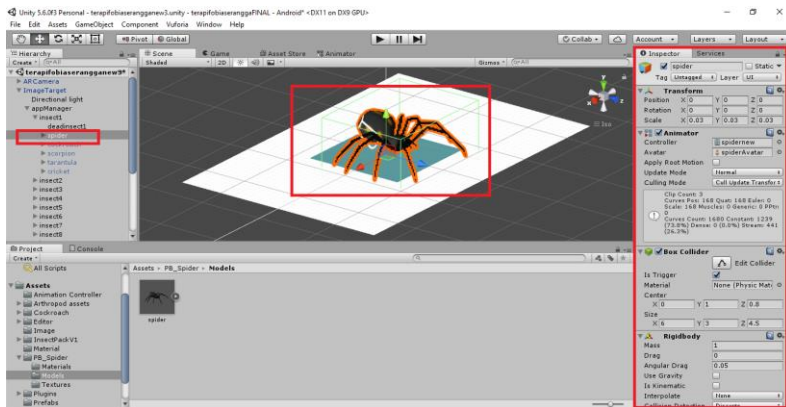
Gambar 4.2 Tampilan *scene* yang telah dibuat



Gambar 4.3 Tampilan *project explorer*



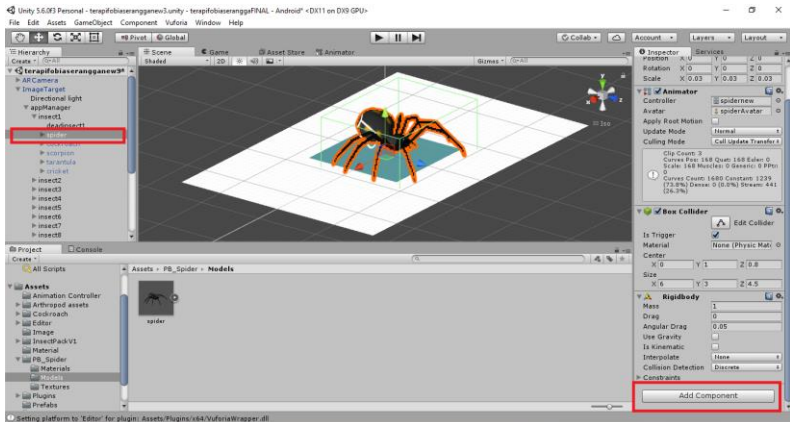
Gambar 4.4 Inseri objek ke dalam proyek



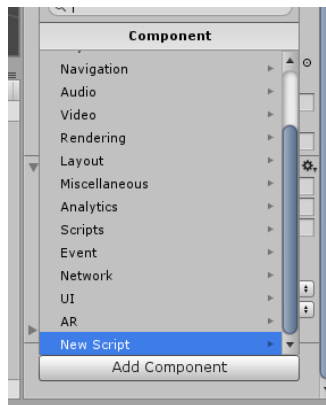
Gambar 4.5 Properti objek

4.2.2 Implementasi Pembuatan Script

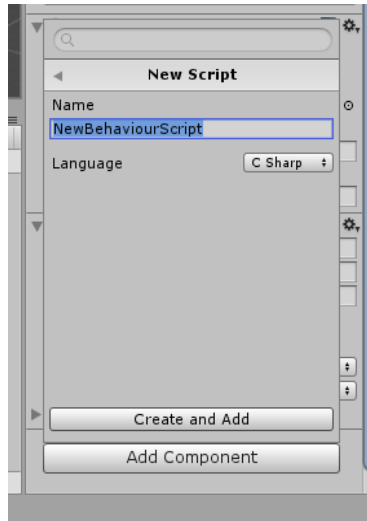
Untuk membuat *script*, dan mengaitkan terhadap objek, pilih objek yang akan dimasukkan *script*, lalu tambahkan komponen *script*. *Script* yang dibuat bisa dalam bentuk *javascript* atau C# (Gambar 4.6, Gambar 4.7 dan Gambar 4.8).



Gambar 4.6 Pembuatan script (1)



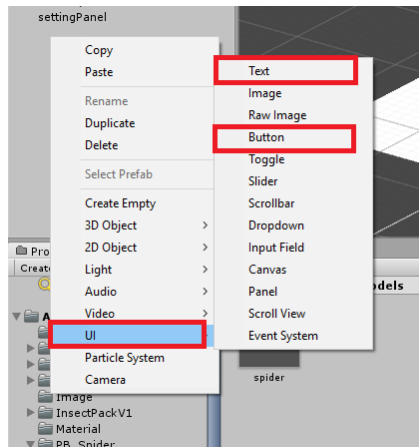
Gambar 4.7 Pembuatan script (2)



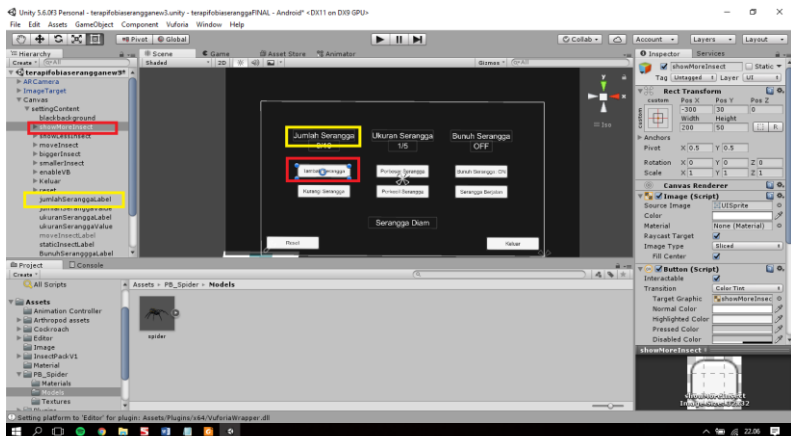
Gambar 4.8 Pembuatan *script* (3)

4.2.3 Implementasi Pembuatan *Canvas*, *Button*, dan *Text*

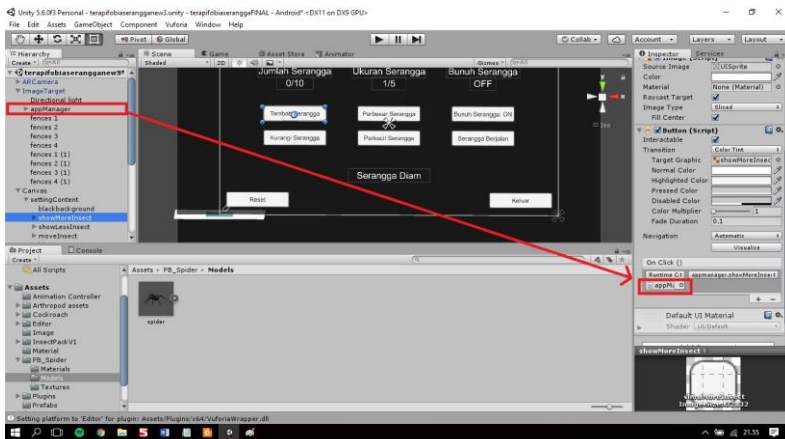
Objek UI (*User Interface*) pada Unity akan selalu menjadi *child* dari objek *canvas*. Jika *canvas* tidak ada pada *parent* objek UI, maka Unity akan langsung membuat sebuah *canvas* baru sebagai *parent* dari objek tersebut. Untuk menambah objek UI *button* dan *text*, gunakan klik kanan pada *project hierarchy* lalu pilih UI, seperti pada Gambar 4.9, lalu pilih *button* atau *text*. *Button* dan *text* sudah berhasil dibuat ditunjukkan pada Gambar 4.10. *Text* berfungsi untuk memberikan label pada objek, sedangkan *button* secara *default* memiliki fungsi *On Click*. Fungsi ini membutuhkan *Game Object* yang memiliki *script*. Penambahan objek *game* pada fungsi *On Click* pada Gambar 4.11 dan pemilihan fungsi pada Gambar 4.12.



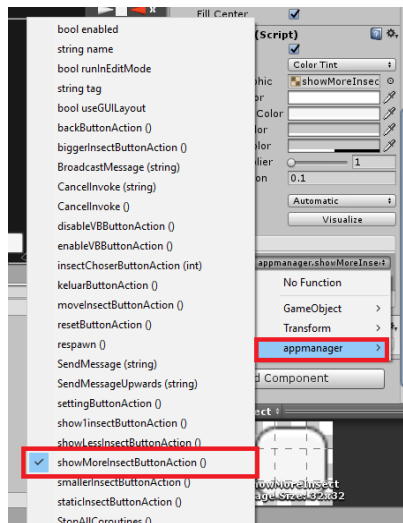
Gambar 4.9 Pembuatan *button* dan *text*



Gambar 4.10 Tampilan *button* dan *text* yang sudah dibuat



Gambar 4.11 Penambahan objek *game* pada fungsi *on click*

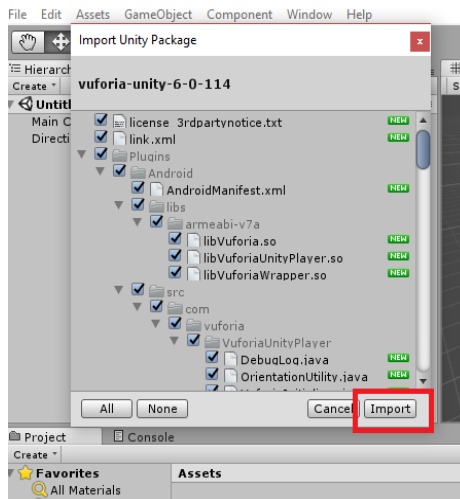


Gambar 4.12 Penambahan fungsi pada fungsi *on click*

4.4.4 Implementasi Penambahan Serangga

Untuk menampilkan serangga ke dunia nyata secara *Augmented Reality*, kita harus mengimpor *package* Vuforia ke *project* Unity. *Package* tersebut bisa diunduh di halaman resmi Vuforia dengan alamat <https://developer.vuforia.com/downloads/sdk>. Cara mengimpor *package* tersebut cukup klik dua kali pada file yang telah diunduh. Namun pastikan aplikasi Unity sudah berjalan dahulu. Cara mengimpor *package* Vuforia tampak pada Gambar 4.13.

Script appmanager digunakan untuk menambahkan serangga yang ditampilkan. Gambar 4. 14 menunjukkan potongan *script* untuk menambah serangga.



Gambar 4.13 Impor package Vuforia

```

1 public GameObject [] insect;
2 int countInsect, maxInsect;
3 ...
4 public void showMoreInsect(int insectChoser){
5     if (countInsect < maxInsect) {
6         if (!insect[0].activeSelf) {
7             insect[0].gameObject.SetActive (true);
8             if (insectChoser != -1) {
9                 insect[0].transform.GetChild
10                    (insectChoser).gameObject.SetActive
11                    (true);
12             }
13             spawnrot = new Vector3 (0, Random.Range
14                (0, 360), 0);
15             insect[0].gameObject.transform.Rotate
16                (spawnrot);
17         } else {
18             for (int f = 1; f < 10; f++) {
19                 if (!insect [f].activeSelf) {
20                     while (!insect [f].activeSelf) {
21                         Vector3 spawnpos = new Vector3
22                            (Random.Range (-8, 8), 0,
23                             Random.Range (-8, 8));
24                         if (Physics.OverlapBox (spawnpos *
25                            100, spawnsize).Length == 0) {
26                             insect [f].gameObject.SetActive
27                                (true);
28                             if (insectChoser != -1) {
29                                 insect [f].transform.GetChild
30                                    (insectChoser).gameObject.
31                                    SetActive(true);
32                             }
33                             insect[f].gameObject.transform.
34                                localPosition = spawnpos / 10;
35                             spawnrot = new Vector3 (0,
36                                Random.Range (0, 360), 0);
37                             insect [f].gameObject.transform.
38                                Rotate (spawnrot);
39                         }
40                     }
41                     break;
42                 }
43             }
44         }
45         countInsect++;
46         jumlahSeranggaValue.GetComponent<Text>
47            ().text = countInsect.ToString() + "/10";
48         jumlahSeranggaValue2.GetComponent<Text>
49            ().text = countInsect.ToString() + "/10";

```

```

50     }
51     if (moveinsectstat == 1) {
52         moveInsect ();
53     }
54     if (moveinsectstat == 0) {
55         staticInsect ();
56     }
57 }
58 ...

```

Gambar 4.14 Script menambah serangga

4.2 Implementasi Mengurangi Serangga

Implementasi mengurangi serangga dilakukan dengan menggunakan fungsi yang terdapat pada *script appmanager* seperti yang ditampilkan pada Gambar 4.15.

```

1 public GameObject [] insect;
2 int countInsect;
3 ...
4 void showlessinsect(){
5     if (countInsect > 0) {
6         insect[countInsect-1].gameObject.SetActive
7         (false);
8         for (int l = 1; l < 6; l++) {
9             insect[countInsect-1].transform.
10             GetChild(l).gameObject.SetActive (false)
11         }
12         countInsect--;
13         jumlahSeranggaValue.GetComponent<Text>
14         ().text = countInsect.ToString () + "/10";
15         jumlahSeranggaValue2.GetComponent<Text>
16         ().text = countInsect.ToString() + "/10";
17     }
18 }
19 ...

```

Gambar 4.15 Script mengurangi serangga

4.3 Implementasi Menjalankan dan Menghentikan Serangga

Implementasi menjalankan dan menghentikan serangga dibagi menjadi 2 subbab meliputi implementasi serangga berjalan dan diam dan pembuatan sistem *collision*.

4.5.1 Implementasi Serangga Berjalan dan Serangga Diam

Serangga berjalan dibuat dengan menambahkan *script moveInsect* pada masing-masing objek serangga. Gambar 4.16 menunjukkan potongan *script moveInsect*. Untuk menjalankan serangga, digunakan potongan *script* yang terdapat di *appmanager* seperti yang ditunjukkan pada Gambar 4.17. Untuk menghentikan serangga, digunakan potongan *script* yang terdapat di *appmanager* seperti yang ditunjukkan pada Gambar 4.18.

1	<code>public float speed;</code>
2	<code>void Update () {</code>
3	<code> Vector3 pos = gameObject.transform.localPosition;</code>
4	<code> transform.Translate (Vector3.forward * speed *</code>
5	<code> Time.deltaTime);</code>
6	<code> ...</code>
7	<code>}</code>

Gambar 4.16 Script moveInsect

1	<code>public GameObject [] insect;</code>
2	<code>private Animator [] spideranim, cockroachanim,</code>
3	<code>scorpioanim, tarantulaanim, cricketanim;</code>
4	<code>...</code>
5	<code>void moveInsect(){</code>
6	<code> staticInsectLabel.gameObject.SetActive (false);</code>
7	<code> moveInsectLabel.gameObject.SetActive (true);</code>
8	<code> moveinsectstat = 1;</code>
9	<code> for (int j = 0; j < 10; j++) {</code>
10	<code> (insect[j].GetComponent ("moveInsect") as</code>
11	<code> MonoBehaviour).enabled = true;</code>
12	<code> spideranim[j].SetInteger ("insectstate", 1);</code>
13	<code> cockroachanim[j].SetInteger ("insectstate",</code>
14	<code> 1);</code>
15	<code> scorpioanim[j].SetInteger ("insectstate", 1);</code>
16	<code> tarantulaanim[j].SetInteger ("insectstate",</code>
17	<code> 1);</code>
18	<code> cricketanim[j].SetInteger ("insectstate", 1);</code>
19	<code> }</code>
20	<code>}</code>
21	<code>...</code>

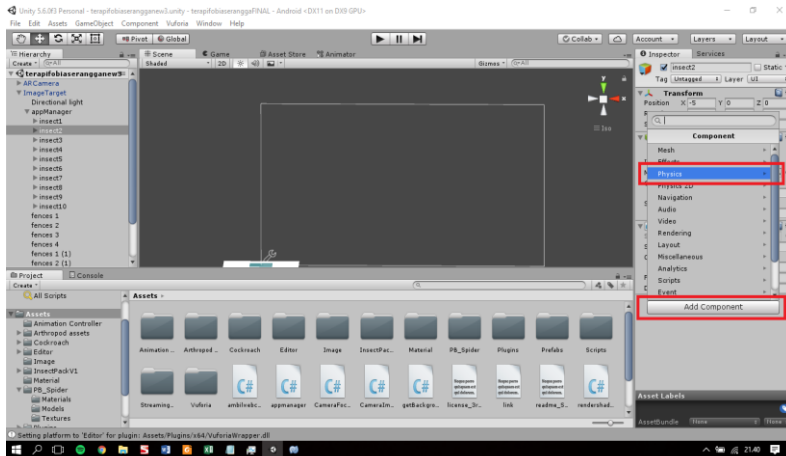
Gambar 4.17 Script menjalankan serangga

1	public GameObject [] insect;
2	private Animator [] spideranim, cockroachanim,
3	scorpioanim, tarantulaanim, cricketanim;
4	...
5	void staticInsect(){
6	staticInsectLabel.gameObject.SetActive (true);
7	moveInsectLabel.gameObject.SetActive (false);
8	moveinsectstat = 0;
9	for (int j = 0; j < 10; j++) {
10	(insect[j].GetComponent ("moveInsect") as
11	MonoBehaviour).enabled = false;
12	spideranim[j].SetInteger ("insectstate", 0);
13	cockroachanim[j].SetInteger ("insectstate",
14	0);
15	scorpioanim[j].SetInteger ("insectstate", 0);
16	tarantulaanim[j].SetInteger ("insectstate",
17	0);
18	cricketanim[j].SetInteger ("insectstate", 0);
19	}
20	}
21	...

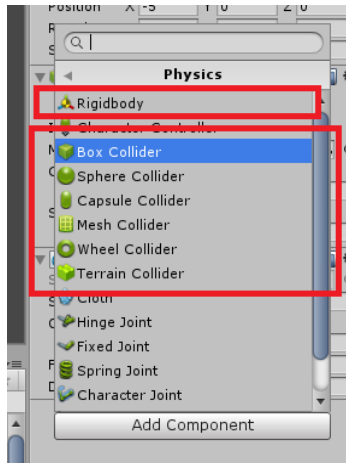
Gambar 4.18 Script menghentikan serangan

4.5.2 Implementasi Sistem Collision

Langkah pertama yang perlu dilakukan untuk membuat *collision* pada serangan adalah dengan menambahkan komponen *collider* ke objek serangan dengan ukuran sesuai besar objek serangan. Selanjutnya tambahkan komponen *rigidbody* pada objek serangan. Penambahan *collider* dan *rigidbody* ditunjukkan pada Gambar 4.19 dan Gambar 4.20. Kemudian *script* ditambahkan untuk merotasi serangan ketika mengalami *collision*. Potongan *script* untuk *collision* ditunjukkan pada Gambar 4.21. Agar *script* dapat berjalan saat terjadi *collision*, centang kolom “Is Trigger” pada komponen *collider* seperti yang ditunjukkan Gambar 4.22.



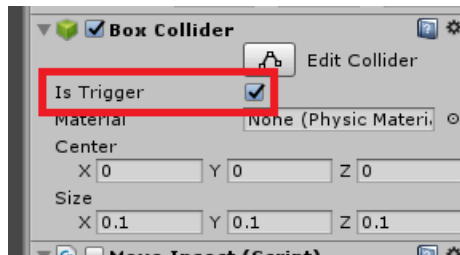
Gambar 4.19 Penambahan Collider dan Rigidbody



Gambar 4.20 Penambahan Collider dan Rigidbody

1	...
2	void OnTriggerEnter(Collider other){
3	Vector3 rotasi = new Vector3 (0, 210, 0);
4	gameObject.transform.Rotate (rotasi);
5	}
6	...

Gambar 4.21 Script Collision



Gambar 4.22 Kolom “Is Trigger”

4.4 Implementasi Batas Pergerakan Serangga

Implementasi batas pergerakan serangga dilakukan dengan menggunakan fungsi yang terdapat pada *script moveInsect* seperti yang ditampilkan Gambar 4.23.

1	void Update () {
2	if (gameObject.transform.localPosition.x < -1
3	gameObject.transform.localPosition.x > 1
4	gameObject.transform.localPosition.z < -1
5	gameObject.transform.localPosition.z > 1) {
6	Vector3 rotasi = new Vector3 (0, 90, 0);
7	gameObject.transform.Rotate (rotasi);
8	}
9	}

Gambar 4.23 Script pembatasan gerakan serangga

4.5 Implementasi Memperbesar Serangga

Implementasi memperbesar serangga dilakukan dengan menggunakan fungsi yang terdapat pada *script appmanager* seperti yang ditampilkan Gambar 4.24.

1	public GameObject [] insect;
2	float insectSize, maxsizeinsect, scaleupvalue;
3	int insectSizeShown;
4	...
5	void biggerInsect(){
6	if (insectSize < maxsizeinsect) {
7	insectSize += scaleupvalue;
8	insectSizeShown++;
9	ukuranSeranggaValue.GetComponent<Text>
10	().text = insectSizeShown.ToString () +
11	"/5";
12	for (int p = 0; p < 9; p++) {
13	insect[p].transform.localScale = new
14	Vector3 (insectSize,
15	insectSize,insectSize);
16	}
17	}
18	}
19	...

Gambar 4.24 Script memperbesar serangga

4.6 Implementasi Memperkecil Serangga

Implementasi memperbesar serangga dilakukan dengan menggunakan fungsi yang terdapat pada script appmanager seperti yang ditampilkan Gambar 4.25.

1	public GameObject [] insect;
2	float insectSize, maxsizeinsect, scaleupvalue;
3	int insectSizeShowned;
4	...
5	void smallerInsect(){
6	if (insectSize > minsizeinsect) {
7	insectSize -= scaleupvalue;
8	insectSizeShowned--;
9	ukuranSeranggaValue.GetComponent<Text>
10	().text = insectSizeShowned.ToString () +
11	"/5";
12	for (int p = 0; p < 9; p++) {
13	insect[p].transform.localScale = new
14	Vector3 (insectSize,
15	insectSize,insectSize);
16	}
17	}
18	}
19	...
20	
21	

Gambar 4.25 Script memperkecil serangga

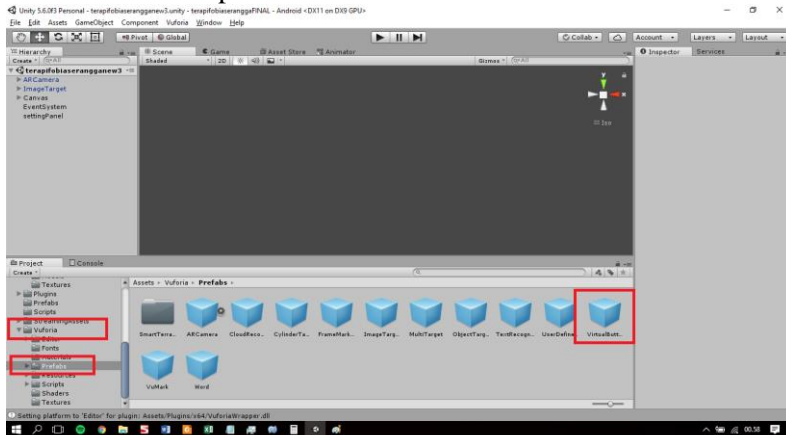
4.7 Implementasi Fitur Bunuh Serangga

Implementasi fitur bunuh serangga dibagi menjadi 3 subbab meliputi pembuatan *virtual button*, pengaktifan dan penonaktifan fitur bunuh serangga, dan implementasi *respawn* serangga.

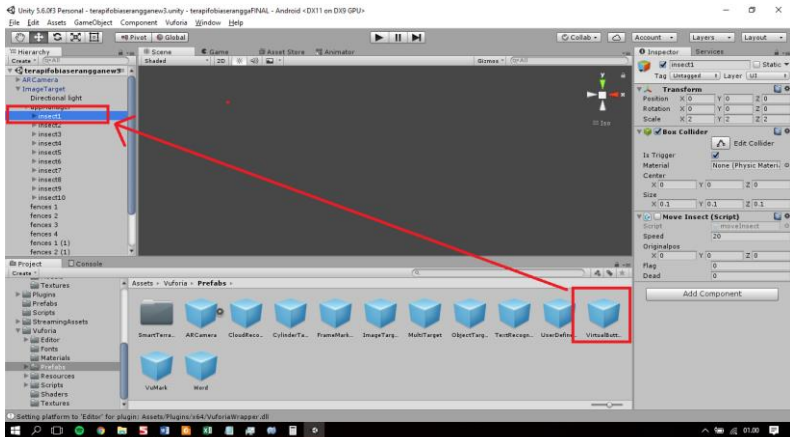
4.7.1 Implementasi *Virtual Button*

Virtual button merupakan fitur dari Vuforia yang berupa tombol virtual yang dapat ditekan di atas *marker*. Untuk menambahkan *virtual button*, buka direktori vuforia/Prefabs pada *project explorer*. Kemudian *drag* prefab *virtual button* ke objek serangga yang akan kita tambahkan *virtual button*. Pastikan menambahkan *virtual button* pada objek yang merupakan *child* dari *ImageTarget*. Setelah itu atur ukuran *virtual button* sesuai dengan ukuran serangga dan lakukan pengaturan pada komponen *virtual button* sesuai kebutuhan. Untuk lebih jelasnya, penambahan *virtual button* ditunjukkan pada Gambar 4.26, Gambar 4.27, dan Gambar 4.28.

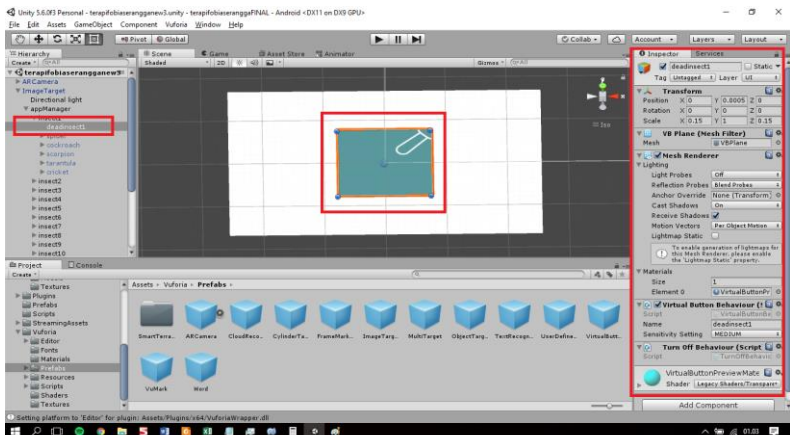
Untuk menambahkan fungsi jika *virtual button* ditekan, tambahkan *script* pada *ImageTarget* dengan nama *VBEventHandler*. Gambar 4.29 menunjukkan penambahan *script* *VBEventHandler*. Gambar 4.30 menunjukkan fungsi saat *virtual button* ditekan. Gambar 4.31 dan 4.32 menunjukkan fungsi saat *virtual button* dilepas.



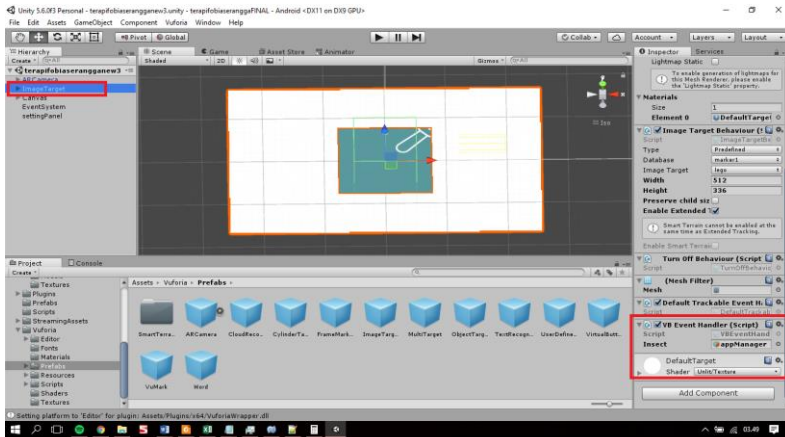
Gambar 4.26 Penambahan *virtual button*



Gambar 4.27 Penambahan *virtual button* ke objek game



Gambar 4.28 *Virtual button* yang telah berhasil ditambahkan



Gambar 4.29 Penambahan *script VBEventHandler*

1	public GameObject insect;
2	...
3	public void
4	OnButtonPressed(VirtualButtonAbstractBehaviour vb){
5	if (vb!= null) {
6	switch (vb.VirtualButtonName) {
7	case "deadinsect1":
8	insect.transform.GetChild(0).transform.GetChild(1).Ge
9	tComponent<Animator> ().SetInteger ("insectstate",
10	2);
11	insect.transform.GetChild(0).transform.GetChild(2).Ge
12	tComponent<Animator> ().SetInteger ("insectstate",
13	2);
14	insect.transform.GetChild(0).transform.GetChild(3).Ge
15	tComponent<Animator> ().SetInteger ("insectstate",
16	2);
17	insect.transform.GetChild(0).transform.GetChild(4).Ge
18	tComponent<Animator> ().SetInteger ("insectstate",
19	2);
20	insect.transform.GetChild(0).transform.GetChild(5).Ge
21	tComponent<Animator> ().SetInteger ("insectstate",
22	2);
23	(insect.transform.GetChild(0).GetComponent("moveInsec
24	t") as MonoBehaviour).enabled = false;
25	break;
26	...
27	}
28	}
29	}

Gambar 4.30 Script VBEventHandler saat virtual button ditekan

1	public GameObject insect;
2	...
3	public void
4	OnButtonReleased(VirtualButtonAbstractBehaviour vb){
5	if (vb!=null) {
6	switch (vb.VirtualButtonName) {
7	case "deadinsect1":
8	StartCoroutine (deadAction (0));
9	break;
10	...
11	}
12	}
13	}

Gambar 4.31 Script VBEventHandler saat virtual button dilepas

1	public GameObject insect;
2	...
3	IEnumerator deadAction(int insectnum) {
4	yield return new WaitForSeconds(1);
5	insect.transform.GetChild(insectnum).gameObject.
6	GetComponent<moveInsect> ().dead = 1;
7	insect.transform.GetChild(insectnum).gameObject.
8	SetActive(false);
9	insect.GetComponent<appmanager> ().respawn ();
10	}

Gambar 4.32 Script VBEventHandler saat virtual button dilepas (2)

4.7.2 Implementasi Pengaktifan dan Penonaktifan Fitur Bunuh Serangga

Implementasi pengaktifan dan penonaktifan fitur bunuh serangga dilakukan dengan menggunakan fungsi yang terdapat pada *script* appmanager. Fungsi pengaktifan fitur bunuh serangga ditunjukkan pada Gambar 4.33. Fungsi penonaktifan fitur bunuh serangga ditunjukkan pada Gambar 4.34.

1	public GameObject [] insect;
2	...
3	public void enableVB(){
4	bunuhSeranggaValue.GetComponent<Text>().text =
5	"ON";
6	for (int g = 0; g < 10; g++) {
7	(insect[g].transform.GetChild(0).gameObject.
8	GetComponent("VirtualButtonBehaviour") as
9	MonoBehaviour).enabled = true;
10	}
11	}
12	...
13	

Gambar 4.33 Script mengaktifkan fitur bunuh serangga

1	public GameObject insect1, insect2, insect3, insect4,
2	insect5, insect6, insect7, insect8, insect9, insect10;
3	...
4	public void disableVB(){
5	bunuhSeranggaValue.GetComponent<Text>().text =
6	"OFF";
7	for (int g = 0; g < 10; g++) {
8	(insect[g].transform.GetChild(0).gameObject.
9	GetComponent("VirtualButtonBehaviour") as
10	MonoBehaviour).enabled = false;
11	}
12	}
13	...}

Gambar 4.34 Script menonaktifkan fitur bunuh serangga

4.7.3 Implementasi *Respawn* Serangga

Implementasi *respawn* serangga dilakukan dengan menggunakan fungsi yang terdapat pada *script* appmanager seperti yang ditampilkan Gambar 4.35.

```

1 public GameObject[] insect;
2 public int moveinsectstat;
3 ...
4 public void respawn(){
5     if (insect[0].gameObject.GetComponent
6         <moveInsect> ().dead == 1) {
7         insect[0].gameObject.GetComponent<moveInsect>
8             ().dead = 0;
9         while (!insect[0].activeSelf) {
10             Vector3 spawnpos = new Vector3
11                 (Random.Range (-8, 8), 0, Random.Range (-8,
12                     8));
13             if (Physics.OverlapBox (spawnpos * 100,
14                 spawnsize).Length == 0) {
15                 insect[0].gameObject.SetActive (true);
16                 insect[0].gameObject.transform.
17                     localPosition = spawnpos / 10;
18                 spawnrot = new Vector3 (0, Random.Range
19                     (0, 360), 0);
20                 insect[0].gameObject.transform.Rotate
21                     (spawnrot);
22             }
23         }
24         if (moveinsectstat == 1) {
25             moveInsect ();
26         }
27         if (moveinsectstat == 0) {
28             staticInsect ();
29         }
30     } else {
31         for (int r = 1; r < 10; r++) {
32             if (insect[r].gameObject.GetComponent
33                 <moveInsect> ().dead == 1) {
34                 insect[r].gameObject.GetComponent
35                     <moveInsect> ().dead = 0;
36                 countInsect--;
37                 showMoreInsect (-1);
38                 break;
39             }
40         }
41     }
42 }
43 ...

```

Gambar 4.35 Script respawn

4.8 Implementasi Fitur Mengembalikan Posisi Serangga Pertama

Implementasi memperbesar serangga dilakukan dengan menggunakan fungsi yang terdapat pada *script appmanager* seperti yang ditampilkan Gambar 4.36.

1	public GameObject [] insect;
2	...
3	void resetButton(){
4	Vector3 spawnpos = new Vector3 (0, 0, 0);
5	insect[0].gameObject.transform.localPosition =
6	spawnpos;
7	}
8	...

Gambar 4.36 Script mengembalikan posisi serangga pertama

4.9 Implementasi Menambah Skenario

Implementasi menambah skenario dilakukan dengan menggunakan fungsi yang terdapat pada *script scenarioManager* seperti yang ditampilkan Gambar 4.37.

```

1 String [] scenarioTemp;
2 String actionValue, durationValue, insertValue;
3 ...
4 public void addAction(int selectEvent){
5     if (currentAction == actionCount) {
6         actionCount++;
7     }
8     scenarioTemp [currentAction] = null;
9     addInsect [currentAction] = 0;
10    scaleInsect [currentAction] = 0;
11    actionValue = actionDropdown.GetComponent<Dropdown>
12    ().value;//set action
13    durationValue = durationAddScenario.
14    GetComponent<InputField> ().text;//set duration
15    int cekNum;
16    somethingWrong = 0;
17    bool cekNumb = int.TryParse (durationValue, out
18    cekNum);
19    disableAllWarning ();
20    if (durationAddScenario.GetComponent<InputField>
21    ().text.Length != 0 && cekNumb) {
22        durationValue = durationAddScenario.
23        GetComponent<InputField>().text;
24        if (jumlahSeranggaDropdown.activeSelf) {//set
25        value
26        insertValue =
27        jumlahSeranggaDropdown.GetComponent<Dropdown>
28        ().value;
29        if (jumlahSeranggaDropdown.
30        GetComponent<Dropdown> ().value == 0 &&
31        addInsectTotal(actionCount) + 1 <= 10) {
32            insertValue = tambahSeranggaDropdown
33            [0].GetComponent<Dropdown> ().value;
34            addInsect[currentAction]++;
35            insertScenarioTemp (insertValue.ToString
36            ());
37            if(addInsectTotal(actionCount)==-1){
38                somethingWrong = 1;
39                insectWarning.SetActive (true);
40                insectWarning.GetComponent<Text> ().text
41                = "Jumlah Serangga Terlalu Banyak";
42            }
43        } else if
44        (jumlahSeranggaDropdown.GetComponent
45        <Dropdown>().value > 0 && addInsectTotal
46        (actionCount) + jumlahSeranggaDropdown.
47        GetComponent<Dropdown>
48        ().value + 1 <= 10) {
49            for (int p = 0; p <=

```

```

50         jumlahSeranggaDropdown.
51         GetComponent<Dropdown> ().value; p++) {
52             insertValue = tambahSeranggaDropdown
53             [p].GetComponent<Dropdown> ().value;
54             addInsect[currentAction]++;
55             if (p != jumlahSeranggaDropdown.
56             GetComponent<Dropdown> ().value) {
57                 scenarioTemp [currentAction] +=
58                 actionValue.ToString () + "a" +
59                 insertValue.ToString () + "v" + "0d";
60             } else if (p == jumlahSeranggaDropdown.
61             GetComponent<Dropdown> ().value) {
62                 insertScenarioTemp (insertValue.
63                 ToString());
64             }
65         }
66         if(addInsectTotal(actionCount)==-1){
67             somethingWrong = 1;
68             insectWarning.SetActive (true);
69             insectWarning.GetComponent<Text> ().text
70             = "Jumlah Serangga Terlalu Banyak";
71         }
72     } else {
73         somethingWrong = 1;
74         insectWarning.SetActive (true);
75         insectWarning.GetComponent<Text> ().text =
76         "Jumlah Serangga Terlalu Banyak";
77     }
78 } else if (kurangiSeranggaDropdown.activeSelf)
79 {
80     insertValue = kurangiSeranggaDropdown.
81     GetComponent<Dropdown> ().value + 1;
82     if (addInsectTotal(actionCount) - insertValue
83     >= 0) {
84         addInsect[currentAction] -= insertValue;
85         insertScenarioTemp (insertValue.ToString
86         ());
87         if(addInsectTotal(actionCount)==-1){
88             somethingWrong = 1;
89             insectWarning.SetActive (true);
90             insectWarning.GetComponent<Text> ().text
91             = "Jumlah Serangga Terlalu Sedikit";
92         }
93     } else {
94         somethingWrong = 1;
95         insectWarning.SetActive (true);
96         insectWarning.GetComponent<Text> ().text =
97         "Jumlah Serangga Terlalu Sedikit";
98     }

```

```

99         } else if (insectSizeDropdown.activeSelf) {
100             insertValue = insectSizeDropdown.
101             GetComponent<Dropdown> ().value + 1;
102             if (scaleInsectTotal(actionCount) +
103             insertValue < 5) {
104                 scaleInsect[currentAction] += insertValue;
105                 insertScenarioTemp (insertValue.ToString
106                 ());
107                 if(scaleInsectTotal(actionCount)==-1){
108                     somethingWrong = 1;
109                     scaleWarning.SetActive (true);
110                     scaleWarning.GetComponent<Text> ().text =
111                     "Ukuran Serangga Terlalu Besar";
112                 }
113             } else {
114                 somethingWrong = 1;
115                 scaleWarning.SetActive (true);
116                 scaleWarning.GetComponent<Text> ().text =
117                 "Ukuran Serangga Terlalu Besar";
118             }
119         } else if (insectSizeDropdown2.activeSelf) {
120             insertValue = insectSizeDropdown2.
121             GetComponent<Dropdown> ().value + 1;
122             if (scaleInsectTotal(actionCount) -
123             insertValue >= 0) {
124                 scaleInsect[currentAction] -= insertValue;
125                 insertScenarioTemp (insertValue.ToString
126                 ());
127                 if(scaleInsectTotal(actionCount)==-1){
128                     somethingWrong = 1;
129                     scaleWarning.SetActive (true);
130                     scaleWarning.GetComponent<Text> ().text =
131                     "Jumlah Serangga Terlalu Kecil";
132                 }
133             } else {
134                 somethingWrong = 1;
135                 scaleWarning.SetActive (true);
136                 scaleWarning.GetComponent<Text> ().text =
137                 "Jumlah Serangga Terlalu Kecil";
138             }
139         } else {
140             insertScenarioTemp("-");
141         }
142     }
143     if (flagedit == 0) {
144         saveScenario = "scenario" + getScenarioStat
145         ().ToString ();
146     }
147     if (flagedit == 1) {

```



```

148         saveScenario = "scenario" +
149             indexedit.ToString();
150     }
151     if (selectEvent == 0 && somethingWrong==0)
152     { //jika pilih save scenario
153         flushAddedScenarioTemp ();
154         PlayerPrefs.SetString (saveScenario,
155             allscenario); //save scenario1, scenario2 dst
156         resetScenario ();
157         backAddScenarioButtonAction ();
158     }
159     else if (selectEvent==1 && somethingWrong==0)
160     { //jika pilih tambah aksi
161         currentAction++;
162         cekActionIndex ();
163         int currentActionPrint = currentAction + 1;
164         actionLabelAddScenario.GetComponent<Text>
165             ().text = "AKSI " + currentActionPrint;
166         defaultformaddscenario ();
167     }
168     } else if (durationAddScenario.GetComponent
169     <InputField>().text.Length == 0) {
170         dataBelumLengkap.SetActive (true);
171     } else if (!cekNumb) {
172         dataWrong.SetActive (true);
173     }
174     if (somethingWrong == 0) {
175         jmlSeranggaValue.GetComponent<Text> ().text =
176             addInsectTotal (actionCount).ToString () +
177             "/10";
178         int insectsizee = scaleInsectTotal
179             (actionCount) + 1;
180         scaleValue.GetComponent<Text> ().text =
181             insectsizee.ToString () + "/5";
182     }
183 }
184 ...

```

Gambar 4.37 Script menambah skenario

4.10 Implementasi Mengubah Skenario

Implementasi mengubah skenario dilakukan dengan menggunakan fungsi yang terdapat pada *script appmanager* seperti yang ditampilkan Gambar 4.38.

```

1  Public GameObjects addScenarioContent,
2  ubahSkenarioContent, dataBelumLengkap, dataWrong;
3  Int flahedit, actionLength;
4  ...
5  public void chooseEditScenario(int indexEdit2){
6      addScenarioContent.SetActive (true);
7      ubahSkenarioContent.SetActive(false);
8      disableAllWarning ();
9      indexEdit2--;
10     string editkey = "scenario" + indexEdit2.ToString
11     ();
12     flagedit = 1;
13     indexedit = indexEdit2;
14     int indexeditprint = indexEdit2 + 1;
15     addScenarioLabel.GetComponent<Text> ().text =
16     "Ubah Skenario " + indexeditprint.ToString();
17     getEachScenario(PlayerPrefs.GetString(editkey));
18     int fill=0;
19     insertScenario = 0;
20     addInsect [insertScenario] = 0;
21     scaleInsect[insertScenario]= 0;
22     while(fill < actionLength-1){
23         scenarioTemp[insertScenario] += action[fill] +
24         "a" + value[fill] + "v" + duration[fill] +
25         "d";
26         if (action [fill] == "0") {
27             addInsect [insertScenario]++;
28         }
29         else if (action [fill] == "1") {
30             int v = int.Parse (value [fill]);
31             addInsect [insertScenario] -= v;
32         }
33         else if (action [fill] == "4") {
34             int v = int.Parse (value [fill]);
35             scaleInsect [insertScenario] = v;
36         }
37         else if (action [fill] == "5") {
38             int v = int.Parse (value [fill]);
39             scaleInsect [insertScenario] -= v;
40         }
41         if (duration[fill]!="0") {
42             insertScenario++;
43             addInsect [insertScenario] = 0;
44             scaleInsect[insertScenario]= 0;
45         }
46         fill++;
47     }
48     jmlSeranggaValue.GetComponent<Text> ().text =
49     addInsectTotal (insertScenario).ToString () +

```

50	"/10";
51	int insectsizee = scaleInsectTotal
52	(insertScenario) + 1;
53	scaleValue.GetComponent<Text> ().text =
54	insectsizee.ToString () + "/5";
55	scaleInsectTotal (insertScenario);
56	actionCount = insertScenario;
57	cekActionIndex ();
58	getEachScenario (scenarioTemp [0]);
59	loadAction();
60	int tempvals = jumlahSeranggaDropdown.
61	GetComponent<Dropdown> ().value;
62	jumlahSeranggaDropdown.GetComponent<Dropdown>
63	().value = jumlahSeranggaDropdown.
64	GetComponent<Dropdown> ().value + 1;
65	jumlahSeranggaDropdown.GetComponent<Dropdown>
66	().value = tempvals;
67	}
68	
69	Public void addAction(){
70	...
71	if (flagedit == 1) {
72	saveScenario = "scenario" + indexedit.ToString
73	();
74	}
75	...
76	}

Gambar 4.38 Script mengubah skenario

4.11 Implementasi Menghapus Skenario

Implementasi menghapus skenario dilakukan dengan menggunakan fungsi yang terdapat pada *script scenarioManager* seperti yang ditampilkan Gambar 4.39.

1	Public Gameobject scenarioContent;
2	...
3	public void hapusScenarioButtonAction(){
4	hapusSkenarioContent.SetActive (true);
5	scenarioContent.SetActive (false);
6	flagedit = 0;
7	cekSkenarioStatus ();
8	}
9	
10	public void deleteScenario(int indexHapus2){
11	string deletekey = "scenario" +
12	indexHapus2.ToString ();
13	PlayerPrefs.SetString (deletekey, "");
14	PlayerPrefs.DeleteKey (deletekey);
15	cekSkenarioStatus ();
16	}
17	...

Gambar 4.39 Script menghapus skenario

4.12 Implementasi Menjalankan Skenario

Implementasi menjalankan skenario dilakukan dengan menggunakan fungsi yang terdapat pada *script scenarioManager* seperti yang ditampilkan Gambar 4.40.

```

1 String[] action, value, duration;
2 Int count, idx;
3 Public GameObject scenarioAction, scenarioTimer;
4 ...
5     public void loadScenario(int scenarionumber){
6         scenarionumber--;
7         string loadkey = "scenario" +
8             scenarionumber.ToString ();
9         getEachScenario (PlayerPrefs.GetString
10             (loadkey).ToString());
11         runScenario ();
12     }
13
14     public void runScenario(){
15         StopAllCoroutines ();
16         backScenarioButtonAction ();
17         this.gameObject.GetComponent<appmanager>
18             ().keluarButtonAction ();
19         this.gameObject.GetComponent<appmanager>
20             ().resetAll();
21         StartCoroutine (runAction ());
22     }
23
24     IEnumerator runAction() {
25         idx = 0;
26         stopSkenario.SetActive (true);
27         settingButton.SetActive (false);
28         if (isPasien == 0) {
29             scenarioAction.SetActive (true);
30             scenarioTimer.SetActive (true);
31         }
32         while(idx<count){
33             int durationn = int.Parse(duration[idx]);
34             if (action [idx] == "0") { //jika aksi =
35                 menambah serangan
36                 scenarioAction.GetComponent<Text> ().text =
37                     "Menambah Serangga";
38                 int val = int.Parse (value [idx]) + 1;
39                 this.gameObject.GetComponent<appmanager>
40                     ().showMoreInsect (val);
41             } else if (action [idx] == "1") { //jika aksi
42                 = mengurangi serangan
43                 scenarioAction.GetComponent<Text> ().text =
44                     "Mengurangi Serangga";
45                 int val = int.Parse (value [idx]);
46                 for (int n = 0; n < val; n++) {
47                     this.gameObject.GetComponent<appmanager>
48                         ().showLessInsectButtonAction ();
49                 }

```

```

50         } else if (action [idx] == "2") { //jika aksi =
51         serangga berjalan
52         scenarioAction.GetComponent<Text> ().text =
53         "Serangga Berjalan";
54         this.gameObject.GetComponent<appmanager>
55         ().moveInsectButtonAction ();
56         } else if (action [idx] == "3") { //jika aksi
57         = serangga diam
58         scenarioAction.GetComponent<Text> ().text =
59         "Serangga Diam";
60         this.gameObject.GetComponent<appmanager>
61         ().staticInsectButtonAction ();
62         } else if (action [idx] == "4") { //jika aksi =
63         perbesar serangga
64         scenarioAction.GetComponent<Text> ().text =
65         "Perbesar Serangga";
66         int val = int.Parse (value [idx]);
67         for (int n = 0; n < val; n++) {
68             this.gameObject.GetComponent<appmanager>
69             ().biggerInsectButtonAction ();
70         }
71         } else if (action [idx] == "5") { //jika aksi =
72         perkecil serangga
73         scenarioAction.GetComponent<Text> ().text =
74         "Perkecil Serangga";
75         int val = int.Parse (value [idx]);
76         for (int n = 0; n < val; n++) {
77             this.gameObject.GetComponent<appmanager>
78             ().smallerInsectButtonAction ();
79         }
80         } else if (action [idx] == "6") { //jika aksi =
81         bunuh serangga:on
82         scenarioAction.GetComponent<Text> ().text =
83         "Bunuh Serangga: ON";
84         this.gameObject.GetComponent<appmanager>
85         ().enableVB ();
86         } else if (action [idx] == "7") { //jika aksi =
87         bunuh serangga:off
88         scenarioAction.GetComponent<Text> ().text =
89         "Bunuh Serangga: OFF";
90         this.gameObject.GetComponent<appmanager>
91         ().disableVB ();
92         } else if (action [idx] == "8") { //jika aksi =
93         reset posisi serangga pertama
94         scenarioAction.GetComponent<Text> ().text =
95         "Reset Posisi Serangga 1";
96         this.gameObject.GetComponent<appmanager>
97         ().resetButtonAction ();
98     }

```

99	idx++;
100	while (durationnn != 0) {
101	scenarioTimer.GetComponent<Text> ().text =
102	durationnn.ToString();
103	yield return new WaitForSeconds (1);
104	durationnn--;
105	}
106	}
107	stopScenario ();
108	this.gameObject.GetComponent<appmanager>
109	().resetAll();
110	if (isPasien == 1) {
111	scenarioButtonAction ();
112	}
113	}
114	...

Gambar 4.40 Script menjalankan skenario

4.13 Implementasi Menghentikan Skenario

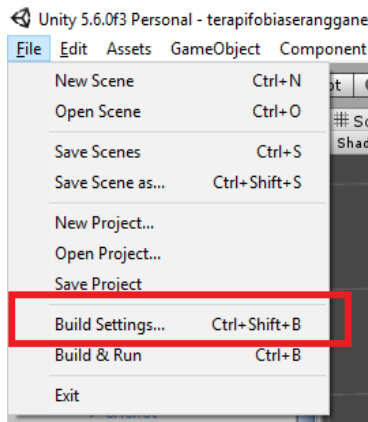
Implementasi menghentikan skenario dilakukan dengan menggunakan fungsi yang terdapat pada *script scenarioManager* seperti yang ditampilkan Gambar 4.41.

1	Public GameObject settingButton, stopSkenario,
2	scenarioAction, scenarioTimer;
3	...
4	public void stopScenario(){
5	StopAllCoroutines ();
6	this.gameObject.GetComponent<appmanager>
7	().resetAll();
8	settingButton.SetActive (true);
9	stopSkenario.SetActive (false);
10	scenarioAction.SetActive (false);
11	scenarioTimer.SetActive (false);
12	}
13	...

Gambar 4.41 Script menghentikan skenario

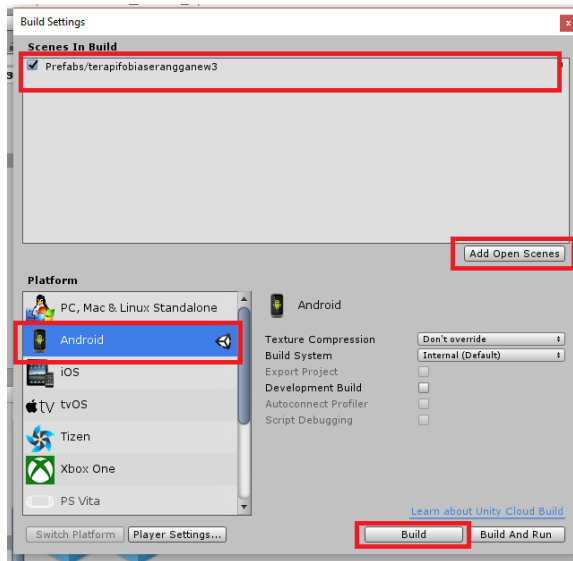
4.14 Pembuatan Proyek

Untuk membuat proyek, hal yang perlu dilakukan adalah memilih *tab* “*file*”, lalu pilih menu “*build settings...*”, proses bisa dilihat pada Gambar 4.42.



Gambar 4.42 Pembuatan proyek

Setelah itu akan muncul menu pembuatan proyek. Pilih mode pembuatan untuk Android. Tambahkan *scene* pada daftar *scenes* yang akan dibuat, klik tombol “*Build*” Gambar 4.43.



Gambar 4.43 Pembuatan proyek (2)

4.15 Cara Menjalankan Aplikasi

Setelah *build* aplikasi berhasil, aplikasi dapat langsung diinstall dan dijalankan di perangkat Android. Aplikasi ini membutuhkan *marker* agar dapat menampilkan objek virtual di dunia nyata. *Marker* harus dicetak pada kertas berukuran A4, kemudian kita arahkan kamera pada perangkat kita kearah *marker* tersebut.

(Halaman ini sengaja dikosongkan)

BAB V

PENGUJIAN DAN EVALUASI

Bab ini membahas hal mengenai uji coba dan evaluasi aplikasi pendukung terapi fobia serangga menggunakan teknologi *Augmented Reality* pada perangkat Android. Uji coba dilakukan menggunakan metode *black box* (kotak hitam) berdasarkan skenario yang telah ditentukan. Uji coba dilakukan pada 1 buah komputer dan 1 buah perangkat Android terhadap hasil implementasi perangkat lunak yang telah dijelaskan pada bab IV.

5.1 Lingkungan Uji Coba

Pada proses uji coba ini, lingkungan dibedakan menjadi lingkungan perangkat komputer dan Android. Berikut ini akan dijelaskan mengenai tiap-tiap lingkungan uji coba aplikasi.

5.1.1 Lingkungan Perangkat Komputer

Lingkungan pelaksanaan uji coba meliputi perangkat keras dan perangkat lunak yang akan digunakan pada sistem ini. Spesifikasi perangkat keras dan perangkat lunak yang digunakan dalam rangka uji coba perangkat lunak ini dicantumkan pada Tabel 5.1.

Tabel 5.1 Lingkungan Perangkat Komputer

No.	Deskripsi
1	Prosesor : Intel® Core™ i5-5200U CPU @ 2.20GHz Memori : 4.00 GB
2	Sistem Operasi Windows 10 Pro 64-bit.

5.1.2 Lingkungan Perangkat Android

Spesifikasi perangkat Android yang digunakan dalam rangka uji coba perangkat lunak ini dicantumkan pada Tabel 5.2

Tabel 5.2 Lingkungan Perangkat Android

No.	Deskripsi
1	Prosesor : Mediatek MT6795 Helio X10 Octa-core 2.0 GHz Cortex-A53
	Memori : 2.00 GB
2	Sistem Operasi Android Versi 5.0 Lollipop

5.2 Pengujian Fungsionalitas

Untuk mengetahui kesesuaian keluaran dari tiap tahap dan langkah penggunaan fitur terhadap skenario yang dipersiapkan, maka dibutuhkan pengujian fungsionalitas. Penjabaran skenario dan hasil uji coba fungsionalitas yang dilakukan terhadap perangkat lunak yang dibangun akan dijabarkan pada subbab ini.

5.2.1 Cara Menjalankan Aplikasi

Aplikasi ini membutuhkan *marker* agar dapat menampilkan objek virtual di dunia nyata. *Marker* harus dicetak pada kertas berukuran A4, kemudian kita arahkan kamera pada perangkat kita kearah *marker* tersebut. Aplikasi ini harus dijalankan di permukaan yang datar dengan pencahayaan yang cukup. Dalam menjalankan aplikasi sebisa mungkin perangkat kamera tidak banyak bergerak agar dapat menampilkan objek virtual dengan stabil.

5.2.2 Skenario Uji Coba Fungsionalitas

Subbab ini akan menjelaskan beberapa skenario uji coba perangkat lunak berdasarkan metode kotak hitam sebagai dasar tolak ukur keberhasilan. Pengujian fungsionalitas terhadap fitur-fitur dari permainan ini akan dijelaskan sebagai berikut:

- Uji coba menambah serangan
- Uji coba mengurangi serangan
- Uji coba menjalankan serangan
- Uji coba menghentikan serangan
- Uji coba memperbesar serangan
- Uji coba memperkecil serangan
- Uji coba mengaktifkan fitur bunuh serangan
- Uji coba menonaktifkan fitur bunuh serangan
- Uji coba mengembalikan posisi serangan pertama
- Uji coba menambah scenario
- Uji coba mengubah scenario
- Uji coba menghapus scenario
- Uji coba menjalankan scenario
- Uji coba menghentikan skenario

Berdasarkan daftar pengujian yang telah disebutkan, dibuat beberapa skenario yang dilakukan pada setiap pengujian tersebut. Penjelasan mengenai cara dan hasil pengujian fungsionalitas perangkat lunak dibahas pada subbab Hasil Uji Coba.

5.2.3 Hasil Uji Coba

Pada subbab ini dijelaskan secara detail mengenai skenario yang dilakukan dan hasil yang didapatkan dari pengujian fungsionalitas perangkat lunak yang dibangun. Penjelasan disajikan dengan menampilkan kondisi awal, masukan, keluaran, hasil yang dicapai, dan kondisi akhir. Berikut ini merupakan

penjabaran skenario dan hasil pengujian yang dicapai pada tiap-tiap fungsionalitas perangkat lunak.

5.2.3.1 Uji Coba Menambah Serangga

Uji coba menambah serangga adalah pengujian fungsi aplikasi dalam menambahkan serangga virtual yang ditampilkan. Uji coba ini bertujuan untuk mengetahui keberhasilan aplikasi dalam menjalankan fungsi menambah serangga. Hasil pengujian secara lengkap dapat dilihat pada Tabel 5.3. Gambar 5.1 menunjukkan tampilan tombol tambah serangga pada menu pengaturan. Gambar 5.2 menunjukkan tampilan pilihan serangga yang dapat ditambahkan. Gambar 5.3 menunjukkan tampilan saat menambahkan serangga. Gambar 5.4 menunjukkan hasil penambahan 1 serangga pada menu pengaturan. Gambar 5.5 menunjukkan hasil penambahan 1 serangga pada tampilan utama. Gambar 5.6 menunjukkan tampilan saat menambahkan banyak serangga. Gambar 5.7 menunjukkan hasil penambahan banyak serangga pada menu pengaturan. Gambar 5.8 menunjukkan hasil penambahan banyak serangga pada tampilan utama.

Tabel 5.3 Hasil uji coba menambah serangga

ID	UJ-UC-001
Nama	Uji Coba Menambah Serangga
Tujuan uji coba	Mengetahui apakah penambahan serangga berhasil dilakukan dengan benar
Kondisi awal	Belum ada serangga yang ditambahkan. Pengguna sudah membuka menu pengaturan
<i>Skenario 1</i>	<i>Pengguna menekan tombol tambah serangga</i>
Masukan	Tekan pada tombol tambah serangga

ID	UJ-UC-001
Keluaran yang diharapkan	Aplikasi akan menampilkan menu tambah serangga yang berisi 5 tombol pilihan serangga yang dapat ditambahkan
Hasil uji coba	Berhasil
Kondisi akhir	Aplikasi menampilkan menu tambah serangga
<i>Skenario 2</i>	<i>Pengguna menambahkan 1 serangga</i>
Masukan	Tekan salah satu dari 5 tombol pilihan serangga pada menu tambah serangga
Keluaran yang diharapkan	Aplikasi akan menampilkan 1 serangga yang dipilih dengan posisi di tengah marker dengan rotasi acak dan jumlah serangga yang ditampilkan pada menu pengaturan dan menu tambah serangga bertambah 1.
Hasil uji coba	Berhasil
Kondisi akhir	Aplikasi menampilkan 1 serangga yang dipilih dengan posisi di tengah marker dan jumlah serangga yang ditampilkan pada menu pengaturan dan menu tambah serangga bertambah 1.
<i>Skenario 3</i>	<i>Pengguna menambahkan serangga lebih dari 1</i>
Masukan	Tekan salah satu dari 5 tombol pilihan serangga pada menu tambah serangga lebih dari 1 kali, maksimal 10 kali. Bisa menekan tombol pilihan serangga yang mana saja dengan aturan setiap kali menekan hanya satu tombol saja yang ditekan.
Keluaran yang diharapkan	Aplikasi akan menampilkan serangga yang pertama ditambahkan berada di tengah marker, dan serangga yang ditambahkan selanjutnya muncul dengan posisi acak dengan rotasi acak. Jumlah serangga yang ditampilkan pada menu pengaturan dan menu tambah serangga akan bertambah setiap kali penambahan serangga

ID	UJ-UC-001
	hingga maksimal 10 serangga. Setelah 10 serangga tidak bisa ditambahkan serangga lagi.
Hasil uji coba	Berhasil
Kondisi akhir	Aplikasi menampilkan serangga pertama dengan posisi di tengah marker dan serangga selanjutnya dengan posisi dan rotasi acak. Jumlah serangga yang ditampilkan pada menu pengaturan dan menu tambah serangga bertambah seiring bertambahnya serangga yang ditampilkan hingga maksimal 10 serangga. Setelah 10 serangga tidak bisa ditambahkan serangga lagi.



Gambar 5.1 Tampilan tombol tambah serangga pada menu pengaturan



Gambar 5.2 Tampilan pilihan serangga yang dapat ditambahkan



Gambar 5.3 Tampilan saat menambahkan serangga



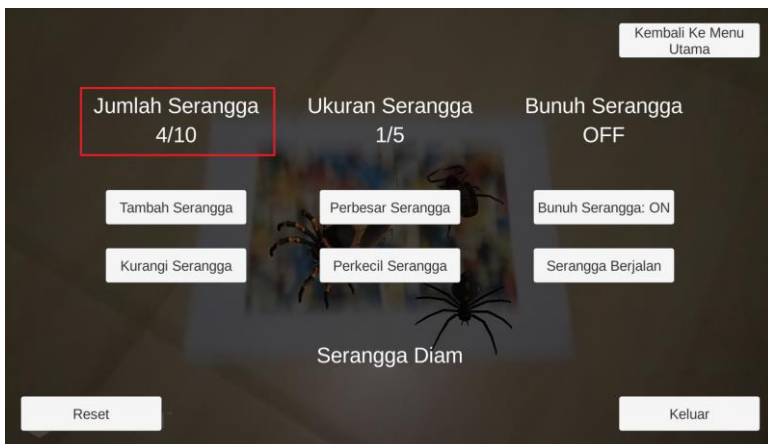
Gambar 5.4 Hasil penambahan 1 serangga pada menu pengaturan



Gambar 5.5 Hasil penambahan 1 serangga pada tampilan utama



Gambar 5.6 Tampilan saat menambahkan banyak serangga



Gambar 5.7 Hasil penambahan banyak serangga pada menu pengaturan



Gambar 5.8 Hasil penambahan banyak serangga pada tampilan utama

5.2.3.2 Uji Coba Mengurangi Serangga

Uji coba mengurangi serangga adalah pengujian fungsi aplikasi dalam mengurangi serangga virtual yang ditampilkan. Uji coba ini bertujuan untuk mengetahui keberhasilan aplikasi dalam menjalankan fungsi mengurangi serangga. Hasil pengujian secara lengkap dapat dilihat pada Tabel 5.4. Gambar 5.9 menunjukkan tampilan sebelum serangga dikurangi pada menu pengaturan. Gambar 5.10 menunjukkan tampilan sebelum serangga dikurangi pada tampilan utama. Gambar 5.11 menunjukkan tampilan saat mengurangi 1 serangga. Gambar 5.12 menunjukkan hasil pengurangan 1 serangga pada tampilan utama.

Tabel 5.4 Hasil uji coba mengurangi serangga

ID	UJ-UC-002
Nama	Uji Coba Mengurangi Serangga
Tujuan uji coba	Mengetahui apakah pengurangan serangga berhasil dilakukan dengan benar
Kondisi awal	Sudah ada serangga yang ditambahkan. Pengguna sudah membuka menu pengaturan
<i>Skenario 1</i>	<i>Pengguna menekan tombol kurangi serangga</i>
Masukan	Tekan pada tombol kurangi serangga
Keluaran yang diharapkan	Aplikasi akan menghilangkan 1 serangga yang terakhir ditambahkan dan jumlah serangga yang ditampilkan pada menu pengaturan dan menu tambah serangga akan berkurang 1 .
Hasil uji coba	Berhasil
Kondisi akhir	Aplikasi menghilangkan 1 serangga yang terakhir ditambahkan dan jumlah serangga yang ditampilkan pada menu pengaturan dan menu tambah serangga berkurang 1.



Gambar 5.9 Tampilan sebelum serangga dikurangi pada menu pengaturan



Gambar 5.10 Tampilan sebelum serangga dikurangi pada tampilan utama



Gambar 5.11 Tampilan saat mengurangi 2 serangga



Gambar 5.12 Hasil pengurangan 2 serangga pada tampilan utama

5.2.3.3 Uji Coba Menjalankan dan Menghentikan Serangga

Uji coba menjalankan dan menghentikan serangga adalah pengujian fungsi aplikasi dalam menjalankan dan menghentikan serangga virtual yang ditampilkan. Uji coba ini bertujuan untuk mengetahui keberhasilan aplikasi dalam menjalankan fungsi menjalankan dan menghentikan serangga. Hasil pengujian secara lengkap dapat dilihat pada Tabel 5.5. Gambar 5.13 menunjukkan tampilan saat menjalankan serangga. Gambar 5.14 menunjukkan hasil menjalankan serangga pada menu pengaturan. Gambar 5.15 menunjukkan hasil menjalankan serangga pada tampilan utama. Gambar 5.16 menunjukkan tampilan saat menghentikan serangga. Gambar 5.17 menunjukkan hasil menghentikan serangga pada menu pengaturan. Gambar 5.18 menunjukkan hasil menghentikan serangga pada tampilan utama.

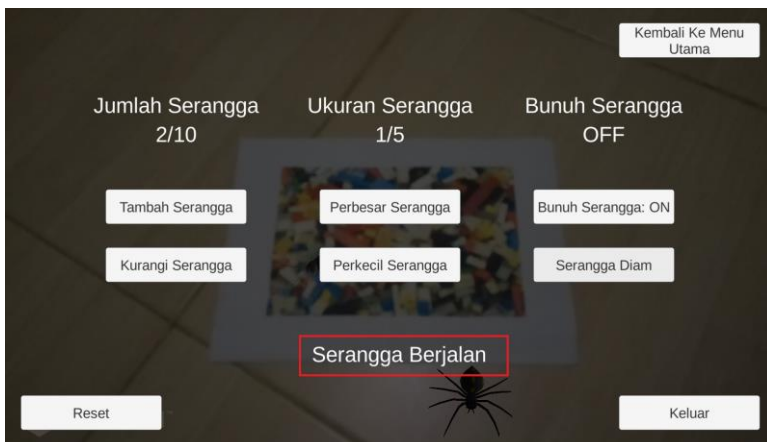
Tabel 5.5 Hasil uji coba menjalankan dan menghentikan serangga

ID	UJ-UC-003
Nama	Uji Coba Menjalankan dan Menghentikan Serangga
Tujuan uji coba	Mengetahui apakah serangga berhasil dijalankan dan dihentikan dengan benar
Kondisi awal	Sudah ada serangga yang ditambahkan Pengguna sudah membuka menu konfigurasi
<i>Skenario 1</i>	<i>Pengguna menjalankan serangga</i>
Masukan	Tekan pada tombol serangga berjalan
Keluaran yang diharapkan	Aplikasi akan menampilkan serangga yang berjalan. Serangga yang saling bertabrakan akan dirotasi 210 derajat. Sedangkan serangga yang berjalan melebihi batas koordinat akan dirotasi 90 derajat. Tombol serangga berjalan akan berubah menjadi tombol serangga diam. Status serangga

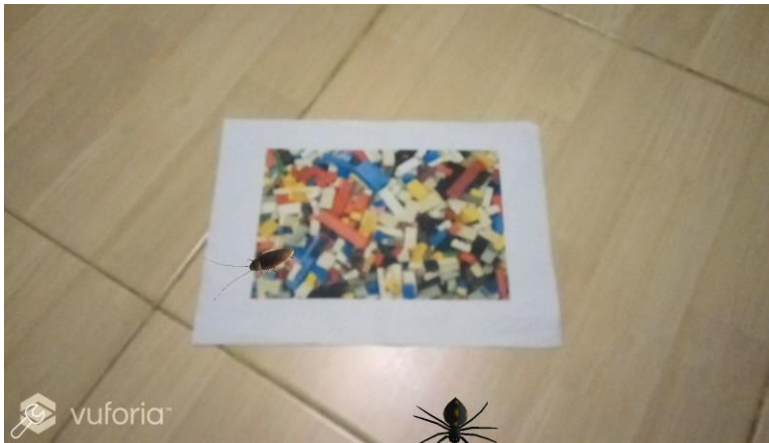
ID	UJ-UC-003
	pada menu pengaturan akan berubah menjadi serangga berjalan
Hasil uji coba	Berhasil
Kondisi akhir	Aplikasi menampilkan serangga yang berjalan. Serangga yang saling bertabrakan dirotasi 210 derajat. Sedangkan serangga yang berjalan melebihi batas koordinat dirotasi 90 derajat. Tombol serangga berjalan berubah menjadi tombol serangga diam. Status serangga pada menu pengaturan berubah menjadi serangga berjalan
<i>Skenario 2</i>	<i>Pengguna menghentikan serangga</i>
Masukan	Tekan pada tombol serangga diam
Keluaran yang diharapkan	Aplikasi akan menampilkan serangga yang diam. Tombol serangga diam akan berubah menjadi tombol serangga berjalan. Status serangga pada menu pengaturan akan berubah menjadi serangga diam
Hasil uji coba	Berhasil
Kondisi akhir	Aplikasi menampilkan serangga yang diam. Tombol serangga diam berubah menjadi tombol serangga berjalan. Status serangga pada menu pengaturan berubah menjadi serangga diam



Gambar 5.13 Tampilan saat menjalankan serangga



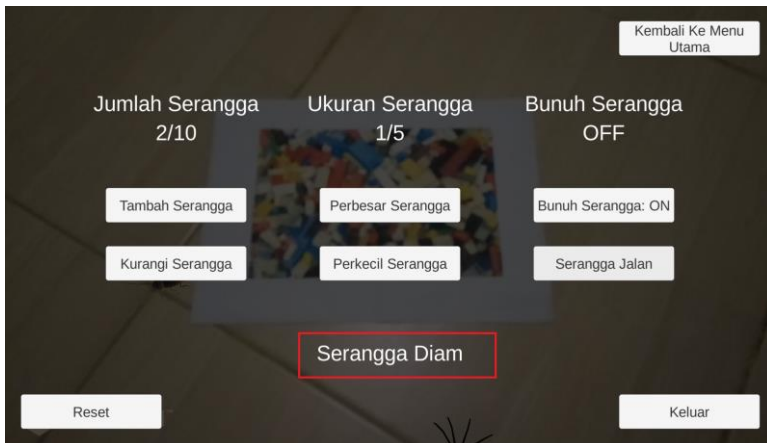
Gambar 5.14 Hasil menjalankan serangga pada menu pengaturan



Gambar 5.15 Hasil menjalankan serangan pada tampilan utama



Gambar 5.16 Tampilan saat menghentikan serangan



Gambar 5.17 Hasil menghentikan serangga pada menu pengaturan



Gambar 5.18 Hasil menghentikan serangga pada tampilan utama

5.2.3.4 Uji Coba Memperbesar Serangga

Uji coba memperbesar serangga adalah pengujian fungsi aplikasi dalam memperbesar serangga virtual yang ditampilkan. Uji coba ini bertujuan untuk mengetahui keberhasilan aplikasi dalam menjalankan fungsi memperbesar serangga. Hasil pengujian secara lengkap dapat dilihat pada Tabel 5.6. Gambar 5.19 menunjukkan tampilan sebelum memperbesar serangga pada menu pengaturan. Gambar 5.20 menunjukkan tampilan sebelum memperbesar serangga pada tampilan utama. Gambar 5.21 menunjukkan tampilan saat memperbesar serangga. Gambar 5.22 menunjukkan hasil memperbesar serangga pada tampilan utama.

Tabel 5.6 Hasil uji coba memperbesar serangga

ID	UJ-UC-004
Nama	Uji Coba Memperbesar Serangga
Tujuan uji coba	Mengetahui apakah serangga berhasil diperbesar dengan benar
Kondisi awal	Sudah ada serangga yang ditambahkan Pengguna sudah membuka menu pengaturan
<i>Skenario 1</i>	<i>Pengguna memperbesar serangga</i>
Masukan	Tekan pada tombol perbesar serangga
Keluaran yang diharapkan	Aplikasi akan menampilkan serangga yang lebih besar. Tampilan skala ukuran serangga pada menu pengaturan akan bertambah 1 setiap kali perbesaran dengan maksimal 4 kali perbesaran. Jika skala ukuran serangga sudah 5, serangga akan tidak dapat diperbesar lagi.
Hasil uji coba	Berhasil
Kondisi akhir	Aplikasi menampilkan serangga yang lebih besar. Tampilan skala ukuran serangga pada menu pengaturan bertambah 1 setiap kali perbesaran

ID	UJ-UC-004
	dengan maksimal 4 kali perbesaran. Jika skala ukuran serangga sudah 5, serangga tidak dapat diperbesar lagi.



Gambar 5.19 Tampilan sebelum memperbesar serangga pada menu pengaturan



Gambar 5.20 Tampilan sebelum memperbesar serangga pada tampilan utama



Gambar 5.21 Tampilan saat memperbesar serangga



Gambar 5.22 Hasil memperbesar serangga pada tampilan utama

5.2.3.5 Uji Coba Memperkecil Serangga

Uji coba memperkecil serangga adalah pengujian fungsi aplikasi dalam memperkecil serangga virtual yang ditampilkan. Uji coba ini bertujuan untuk mengetahui keberhasilan aplikasi dalam menjalankan fungsi memperkecil serangga. Hasil pengujian secara lengkap dapat dilihat pada Tabel 5.7. Gambar 5.23 menunjukkan tampilan sebelum memperkecil serangga pada menu pengaturan. Gambar 5.24 menunjukkan tampilan sebelum memperkecil serangga pada tampilan utama. Gambar 5.25 menunjukkan tampilan saat memperkecil serangga. Gambar 5.26 menunjukkan hasil memperkecil serangga pada tampilan utama.

Tabel 5.7 Hasil uji coba memperbesar serangga

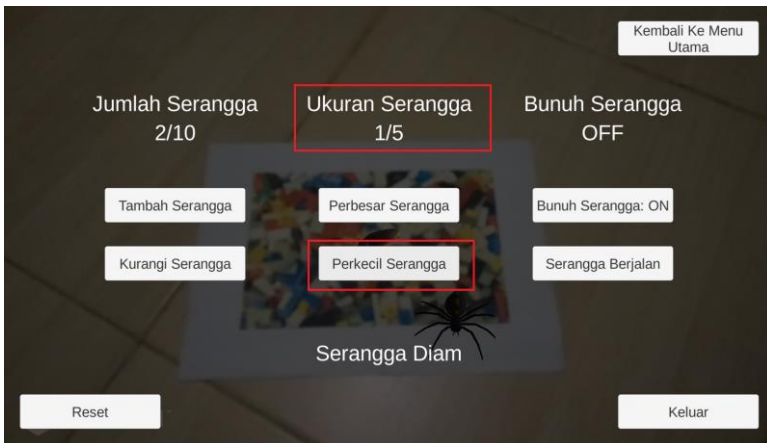
ID	UJ-UC-005
Nama	Uji Coba Memperkecil Serangga
Tujuan uji coba	Mengetahui apakah serangga berhasil diperkecil dengan benar
Kondisi awal	Sudah ada serangga yang ditambahkan dan sudah diperbesar Pengguna sudah masuk menu pengaturan
<i>Skenario 1</i>	<i>Pengguna memperkecil serangga</i>
Masukan	Tekan pada tombol perkecil serangga
Keluaran yang diharapkan	Aplikasi akan menampilkan serangga yang lebih kecil. Tampilan skala ukuran serangga pada menu pengaturan akan berkurang 1 setiap kali pengecilan dengan maksimal 4 kali pengecilan. Jika skala ukuran serangga sudah 1, serangga akan tidak dapat diperkecil lagi.
Hasil uji coba	Berhasil
Kondisi akhir	Aplikasi menampilkan serangga yang lebih kecil. Tampilan skala ukuran serangga pada menu pengaturan berkurang 1 setiap kali pengecilan dengan maksimal 4 kali pengecilan. Jika skala ukuran serangga sudah 1, serangga tidak dapat diperkecil lagi.



Gambar 5.23 Tampilan sebelum memperkecil serangga pada menu pengaturan



Gambar 5.24 Tampilan sebelum memperkecil serangga pada tampilan utama



Gambar 5.25 Tampilan saat memperkecil serangga



Gambar 5.26 Hasil memperkecil serangga pada tampilan utama

5.2.3.6 Uji Coba Fitur Bunuh Serangga

Uji coba fitur bunuh serangga adalah pengujian fungsi aplikasi dalam mengaktifkan dan menonaktifkan fitur bunuh serangga. Uji coba ini bertujuan untuk mengetahui keberhasilan aplikasi dalam menjalankan fungsi mengaktifkan dan menonaktifkan fitur bunuh serangga. Hasil pengujian secara lengkap dapat dilihat pada Tabel 5.8. Gambar 5.27 menunjukkan tampilan saat mengaktifkan fitur bunuh serangga. Gambar 5.28 menunjukkan tampilan setelah mengaktifkan fitur bunuh serangga pada menu pengaturan. Gambar 5.29 menunjukkan uji coba bunuh serangga dengan cara dipukul saat berada di dalam marker. Gambar 5.30 menunjukkan tampilan saat menonaktifkan fitur bunuh serangga. Gambar 5.31 menunjukkan tampilan setelah menonaktifkan fitur bunuh serangga.

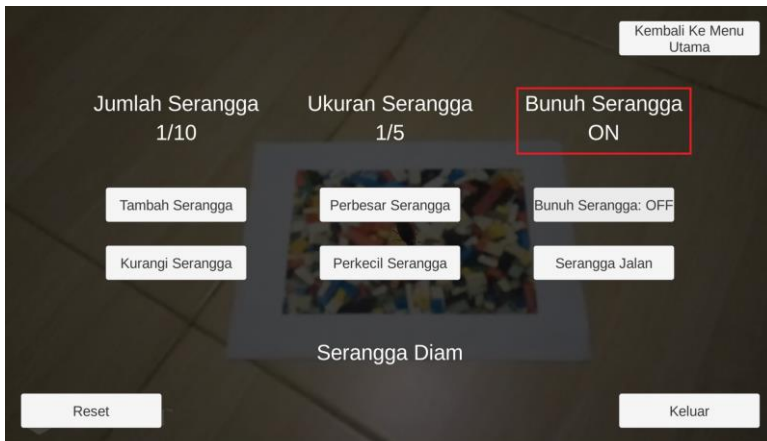
Tabel 5.8 Hasil uji coba mengaktifkan dan menonaktifkan fitur bunuh serangga

ID	UJ-UC-006
Nama	Uji Coba Mengaktifkan dan Menonaktifkan Fitur Bunuh Serangga
Tujuan uji coba	Mengetahui apakah fitur bunuh serangga berhasil diaktifkan dan dinonaktifkan dengan benar
Kondisi awal	Sudah ada serangga yang ditambahkan Pengguna sudah membuka menu pengaturan
<i>Skenario 1</i>	<i>Pengguna mengaktifkan fitur bunuh serangga</i>
Masukan	Tekan pada tombol bunuh serangga: on
Keluaran yang diharapkan	Tombol bunuh serangga: on akan berubah menjadi tombol bunuh serangga: off. Status bunuh serangga pada menu pengaturan akan berubah menjadi bunuh serangga: on. Serangga akan dapat dibunuh dengan cara dipukul selama berada diatas marker.

ID	UJ-UC-006
Hasil uji coba	Berhasil
Kondisi akhir	Tombol bunuh serangga: on berubah menjadi tombol bunuh serangga: off. Status bunuh serangga pada menu pengaturan berubah menjadi bunuh serangga: on. Serangga dapat dibunuh dengan cara dipukul selama berada diatas marker.
<i>Skenario 2</i>	<i>Pengguna menonaktifkan fitur bunuh serangga</i>
Masukan	Tekan pada tombol bunuh serangga: off
Keluaran yang diharapkan	Tombol bunuh serangga: off akan berubah menjadi tombol bunuh serangga: on. Status bunuh serangga pada menu pengaturan akan berubah menjadi bunuh serangga: off. Serangga akan tidak dapat dibunuh.
Hasil uji coba	Berhasil
Kondisi akhir	Tombol bunuh serangga: off berubah menjadi tombol bunuh serangga: on. Status bunuh serangga pada menu pengaturan berubah menjadi bunuh serangga: off. Serangga tidak dapat dibunuh.



Gambar 5.27 Tampilan saat mengaktifkan fitur bunuh serangga



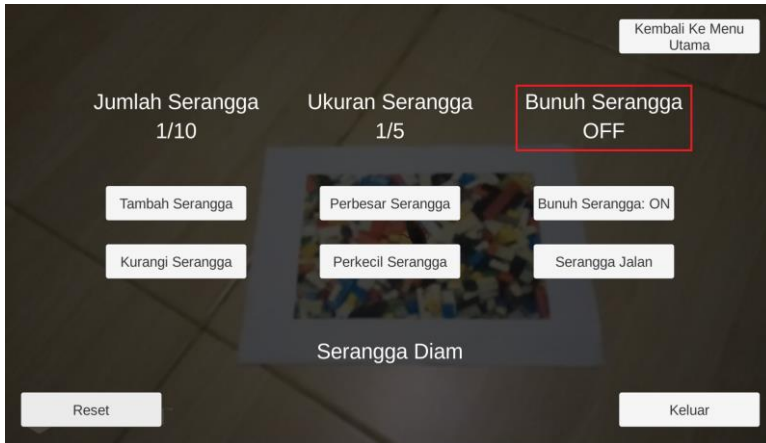
Gambar 5.28 Tampilan setelah mengaktifkan fitur bunuh serangga pada menu pengaturan



Gambar 5.29 Uji coba bunuh serangga dengan cara dipukul saat berada di dalam marker



Gambar 5.30 Tampilan saat menonaktifkan fitur bunuh serangga



Gambar 5.31 Tampilan setelah menonaktifkan fitur bunuh serangga

5.2.3.7 Uji Coba Mengembalikan Posisi Serangga Pertama

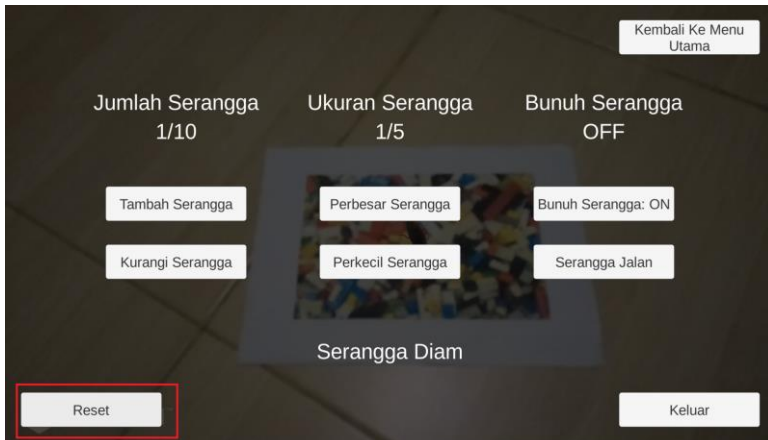
Uji coba mengembalikan posisi serangga pertama adalah pengujian fungsi aplikasi dalam mengembalikan posisi serangga pertama ke tengah *marker*. Uji coba ini bertujuan untuk mengetahui keberhasilan aplikasi dalam menjalankan fungsi mengembalikan posisi serangga pertama. Hasil pengujian secara lengkap dapat dilihat pada Tabel 5.9. Gambar 5.32 menunjukkan tampilan sebelum mengembalikan posisi serangga pertama. Gambar 5.33 menunjukkan tampilan saat mengembalikan posisi serangga pertama. Gambar 5.34 menunjukkan hasil setelah mengembalikan posisi serangga pertama ke tengah *marker*.

Tabel 5.9 Hasil uji coba mengembalikan posisi serangga pertama

ID	UJ-UC-007
Nama	Uji Coba Mengembalikan Posisi Serangga Pertama
Tujuan uji coba	Mengetahui apakah posisi serangga pertama berhasil di kembalikan ke tengah marker
Kondisi awal	Sudah ada serangga yang ditambahkan Pengguna sudah masuk menu pengaturan
<i>Skenario 1</i>	<i>Pengguna mengembalikan posisi serangga pertama</i>
Masukan	Tekan pada tombol <i>reset</i>
Keluaran yang diharapkan	Aplikasi akan menampilkan serangga pertama dengan posisi di tengah marker.
Hasil uji coba	Berhasil
Kondisi akhir	Aplikasi menampilkan serangga pertama dengan posisi di tengah marker



Gambar 5.32 Tampilan sebelum mengembalikan posisi serangga pertama



Gambar 5.33 Tampilan saat mengembalikan posisi serangga pertama



Gambar 5.34 Hasil setelah mengembalikan posisi serangga pertama ke tengah *marker*

5.2.3.8 Uji Coba Menambah Skenario

Uji coba menambah skenario adalah pengujian fungsi aplikasi dalam menambahkan skenario. Uji coba ini bertujuan untuk mengetahui keberhasilan aplikasi dalam menambahkan skenario. Hasil pengujian secara lengkap dapat dilihat pada Tabel 5.10. Gambar 5.35 menunjukkan tampilan tombol kelola skenario pada menu terapis. Gambar 5.36 menunjukkan tombol tambah skenario pada menu skenario. Gambar 5.37 menunjukkan tampilan menu tambah skenario (1). Gambar 5.38 menunjukkan tampilan menu tambah skenario (2). Gambar 5.39 menunjukkan tampilan menu tambah skenario (3).

Tabel 5.10 Hasil uji coba menambahkan skenario

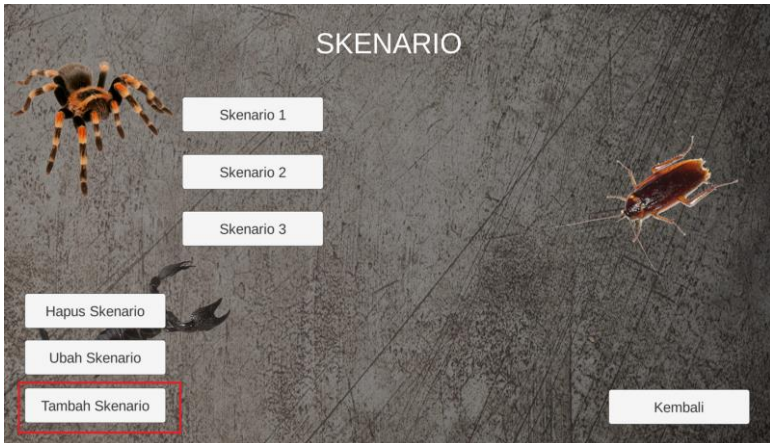
ID	UJ-UC-008
Nama	Uji Coba Menambahkan Skenario
Tujuan uji coba	Mengetahui apakah penambahan skenario berhasil dilakukan dengan benar
Kondisi awal	Jumlah skenario yang ditambahkan belum ada 10 Pengguna sudah masuk menu terapi
<i>Skenario 1</i>	<i>Pengguna menekan tombol kelola skenario</i>
Masukan	Tekan pada tombol <i>skenario</i>
Keluaran yang diharapkan	Aplikasi akan menampilkan menu skenario yang berisi skenario yang tersimpan, tombol tambah skenario, ubah skenario, dan hapus skenario
Hasil uji coba	Berhasil
Kondisi akhir	Aplikasi menampilkan menu skenario yang berisi skenario yang tersimpan, tombol tambah skenario, ubah skenario, dan hapus skenario
<i>Skenario 2</i>	<i>Pengguna menambakan skenario</i>
Masukan	Tekan pada tombol tambah skenario
Keluaran yang diharapkan	Aplikasi akan menampilkan menu tambah skenario yang berisi isian aksi skenario, tombol simpan skenario, dan tombol tambah aksi selanjutnya
Hasil uji coba	Berhasil
Kondisi akhir	Aplikasi akan menampilkan menu tambah skenario yang berisi isian aksi skenario, tombol simpan skenario, dan tombol tambah aksi selanjutnya
<i>Skenario 3</i>	<i>Pengguna menambakan aksi pada skenario</i>

ID	UJ-UC-008
Masukan	Tekan pada tombol tambah aksi selanjutnya
Keluaran yang diharapkan	Aplikasi akan menampilkan isian tambah skenario untuk aksi berikutnya
Hasil uji coba	Berhasil
Kondisi akhir	Aplikasi menampilkan isian tambah skenario untuk aksi berikutnya
<i>Skenario 4</i>	<i>Pengguna melihat aksi sebelumnya</i>
Masukan	Tekan pada tombol aksi sebelumnya
Keluaran yang diharapkan	Aplikasi akan menampilkan isian tambah skenario pada aksi sebelumnya yang berisi data yang tersimpan sebelumnya
Hasil uji coba	Berhasil
Kondisi akhir	Aplikasi menampilkan isian tambah skenario pada aksi sebelumnya yang berisi data yang tersimpan sebelumnya
<i>Skenario 5</i>	<i>Pengguna melihat aksi selanjutnya</i>
Masukan	Tekan pada tombol aksi selanjutnya
Keluaran yang diharapkan	Aplikasi akan menampilkan isian tambah skenario pada aksi selanjutnya yang berisi data yang tersimpan sebelumnya
Hasil uji coba	Berhasil
Kondisi akhir	Aplikasi menampilkan isian tambah skenario pada aksi selanjutnya yang berisi data yang tersimpan sebelumnya
<i>Skenario 6</i>	<i>Pengguna menyimpan skenario</i>

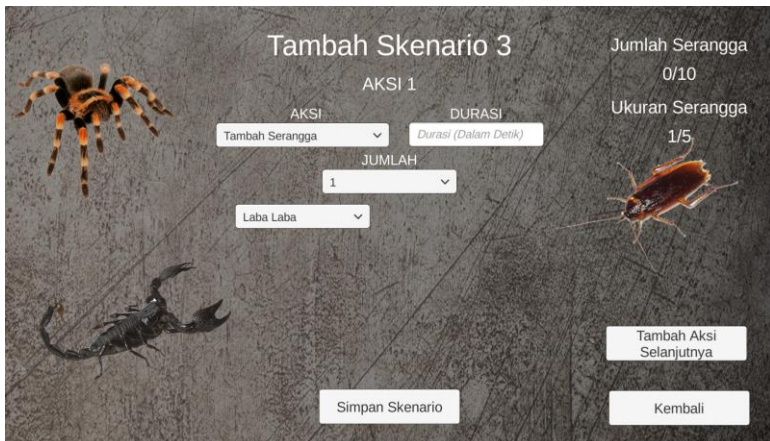
ID	UJ-UC-008
Masukan	Tekan pada tombol simpan skenario
Keluaran yang diharapkan	Aplikasi akan menyimpan skenario dengan nama sesuai dengan nama saat menambahkan skenario
Hasil uji coba	Berhasil
Kondisi akhir	Aplikasi menyimpan skenario dengan nama sesuai dengan nama saat menambahkan skenario



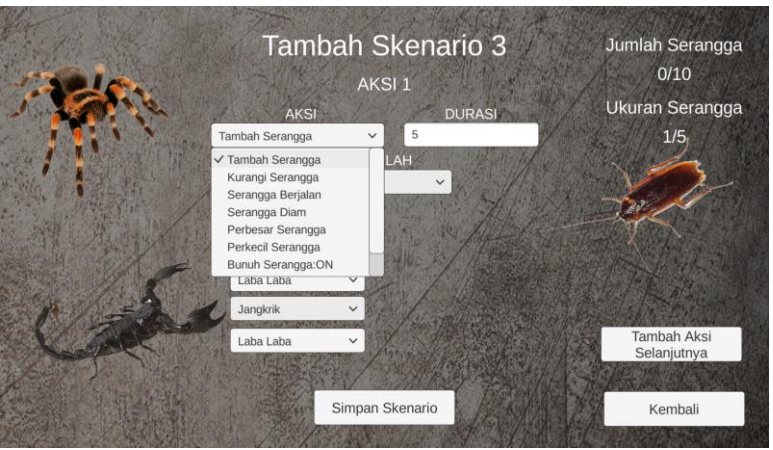
Gambar 5.35 Tampilan tombol kelola skenario pada menu terapi



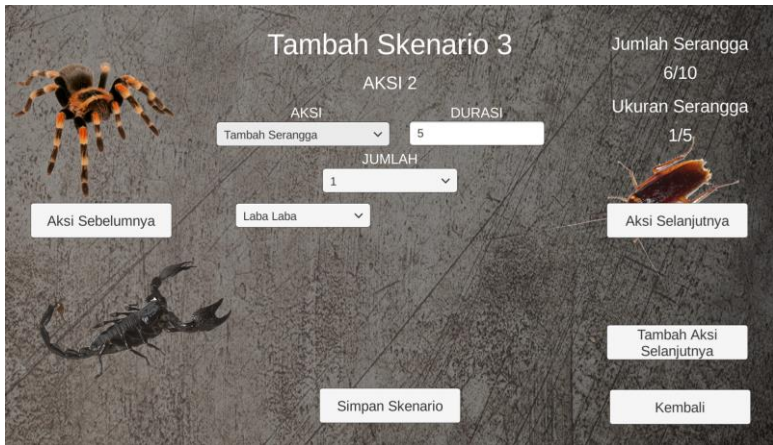
Gambar 5.36 Tampilan menu skenario



Gambar 5.37 Tampilan menu tambah skenario (1)



Gambar 5.38 Tampilan menu tambah skenario (2)



Gambar 5.39 Tampilan menu tambah skenario (3)

5.2.3.9 Uji Coba Mengubah Skenario

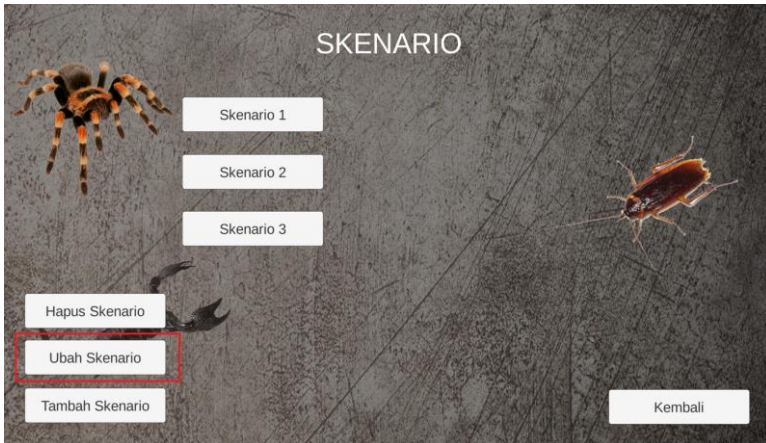
Uji coba mengubah skenario adalah pengujian fungsi aplikasi dalam mengubah skenario yang sudah ditambahkan. Uji coba ini bertujuan untuk mengetahui keberhasilan aplikasi dalam mengubah skenario yang sudah ditambahkan. Hasil pengujian secara lengkap dapat dilihat pada Tabel 5.11. Gambar 5.40 menunjukkan tampilan tombor; ubah skenario pada menu skenario. Gambar 5.41 menunjukkan tampilan pilihan skenario yang dapat diubah. Gambar 5.42 menunjukkan tampilan menu ubah skenario.

Tabel 5.11 Hasil uji coba mengubah skenario

ID	UJ-UC-009
Nama	Uji Coba Mengubah Skenario
Tujuan uji coba	Mengetahui apakah perubahan skenario berhasil dilakukan dengan benar
Kondisi awal	Sudah ada skenario yang ditambahkan Pengguna sudah masuk menu skenario
<i>Skenario 1</i>	<i>Pengguna menekan tombol ubah skenario</i>
Masukan	Tekan pada tombol ubah skenario
Keluaran yang diharapkan	Aplikasi akan menampilkan pilihan skenario yang dapat diubah
Hasil uji coba	Berhasil
Kondisi akhir	Aplikasi menampilkan pilihan skenario yang dapat diubah
<i>Skenario 2</i>	<i>Pengguna merubah skenario</i>
Masukan	Tekan pada tombol pilihan skenario yang dapat diubah
Keluaran yang diharapkan	Aplikasi akan menampilkan menu pengubahan skenario yang berisi isian aksi skenario, tombol

ID	UJ-UC-009
	simpan skenario, tombol aksi selanjutnya, dan tombol tambah aksi selanjutnya.
Hasil uji coba	Berhasil
Kondisi akhir	Aplikasi menampilkan menu pengubahan skenario yang berisi isian aksi skenario, tombol simpan skenario, tombol lihat aksi selanjutnya, dan tombol tambah aksi selanjutnya.
<i>Skenario 3</i>	<i>Pengguna menambahkan aksi pada skenario</i>
Masukan	Tekan pada tombol tambah aksi selanjutnya
Keluaran yang diharapkan	Aplikasi akan menampilkan isian tambah skenario untuk aksi berikutnya
Hasil uji coba	Berhasil
Kondisi akhir	Aplikasi menampilkan isian tambah skenario untuk aksi berikutnya
<i>Skenario 4</i>	<i>Pengguna melihat aksi sebelumnya</i>
Masukan	Tekan pada tombol aksi sebelumnya
Keluaran yang diharapkan	Aplikasi akan menampilkan isian tambah skenario pada aksi sebelumnya yang berisi data yang tersimpan sebelumnya
Hasil uji coba	Berhasil
Kondisi akhir	Aplikasi menampilkan isian tambah skenario pada aksi sebelumnya yang berisi data yang tersimpan sebelumnya
<i>Skenario 5</i>	<i>Pengguna melihat aksi selanjutnya</i>
Masukan	Tekan pada tombol aksi selanjutnya

ID	UJ-UC-009
Keluaran yang diharapkan	Aplikasi akan menampilkan isian tambah skenario pada aksi selanjutnya yang berisi data yang tersimpan sebelumnya
Hasil uji coba	Berhasil
Kondisi akhir	Aplikasi menampilkan isian tambah skenario pada aksi selanjutnya yang berisi data yang tersimpan sebelumnya
<i>Skenario 6</i>	<i>Pengguna menyimpan skenario</i>
Masukan	Tekan pada tombol simpan skenario
Keluaran yang diharapkan	Aplikasi akan menyimpan skenario dengan nama sesuai dengan nama saat menambahkan skenario
Hasil uji coba	Berhasil
Kondisi akhir	Aplikasi menyimpan skenario dengan nama sesuai dengan nama saat menambahkan skenario



Gambar 5.40 Tampilan tombol ubah skenario pada menu skenario



Gambar 5.41 Tampilan pilihan skenario yang dapat diubah

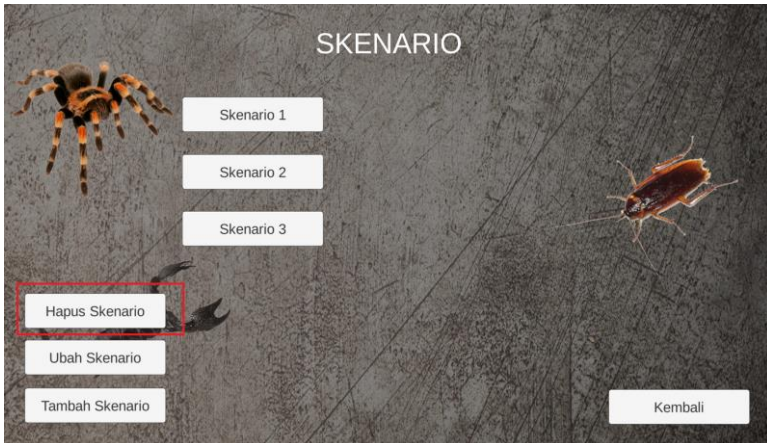
Gambar 5.42 Tampilan menu ubah skenario

5.2.3.10 Uji Coba Menghapus Skenario

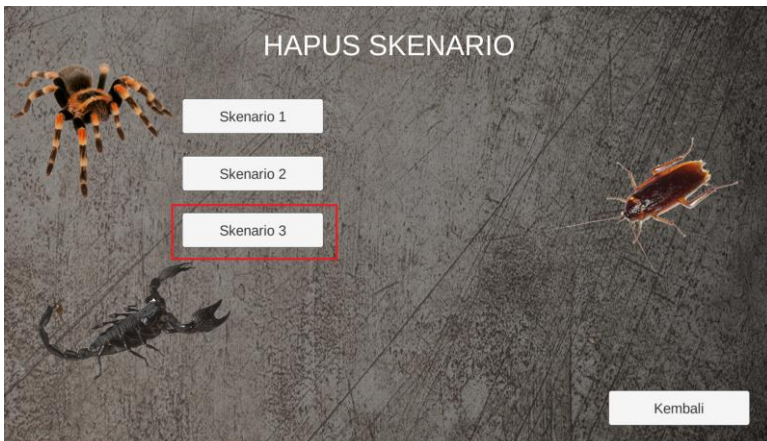
Uji coba menghapus skenario adalah pengujian fungsi aplikasi dalam menghapus skenario sudah ditambahkan. Uji coba ini bertujuan untuk mengetahui keberhasilan aplikasi dalam menghapus skenario yang sudah ditambahkan. Hasil pengujian secara lengkap dapat dilihat pada Tabel 5.12. Gambar 5.43 menunjukkan tampilan tombol hapus skenario pada menu skenario. Gambar 5.44 menunjukkan tampilan pilihan skenario yang dapat dihapus. Gambar 5.45 menunjukkan tampilan setelah menghapus skenario 2.

Tabel 5.12 Hasil uji coba menghapus skenario

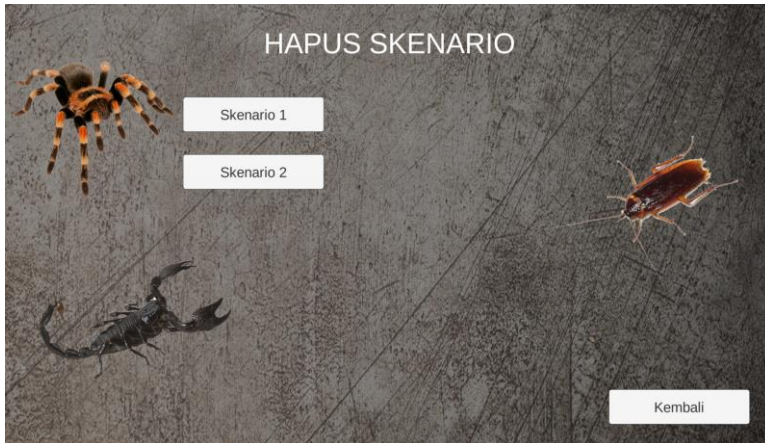
ID	UJ-UC-010
Nama	Uji Coba Menghapus Skenario
Tujuan uji coba	Mengetahui apakah penghapusan skenario berhasil dilakukan dengan benar
Kondisi awal	Sudah ada skenario yang ditambahkan Pengguna sudah masuk menu skenario
<i>Skenario 1</i>	<i>Pengguna menekan tombol hapus skenario</i>
Masukan	Tekan pada tombol hapus skenario
Keluaran yang diharapkan	Aplikasi akan menampilkan pilihan skenario yang dapat dihapus
Hasil uji coba	Berhasil
Kondisi akhir	Aplikasi menampilkan pilihan skenario yang dapat dihapus
<i>Skenario 2</i>	<i>Pengguna menghapus skenario</i>
Masukan	Tekan pada tombol pilihan skenario yang akan dihapus
Keluaran yang diharapkan	Aplikasi akan menghapus skenario. Tombol pilihan skenario yang dihapus akan menghilang.
Hasil uji coba	Berhasil
Kondisi akhir	Aplikasi menghapus skenario. Tombol pilihan skenario yang dihapus menghilang.



Gambar 5.43 Tampilan tombol hapus skenario pada menu skenario



Gambar 5.44 Tampilan pilihan skenario yang dapat dihapus



Gambar 5.45 Tampilan setelah menghapus skenario 2

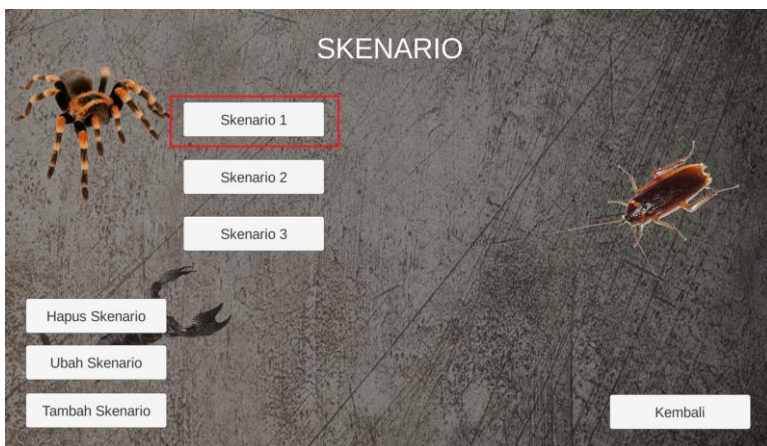
5.2.3.11 Uji Coba Menjalankan dan Menghentikan Skenario

Uji coba menjalankan dan menghentikan skenario adalah pengujian fungsi aplikasi dalam menjalankan dan menghentikan skenario sudah ditambahkan. Uji coba ini bertujuan untuk mengetahui keberhasilan aplikasi dalam menjalankan dan menghentikan skenario yang sudah ditambahkan. Hasil pengujian secara lengkap dapat dilihat pada Tabel 5.13. Gambar 5.46 menunjukkan tampilan pilihan skenario yang dapat dijalankan. Gambar 5.47 menunjukkan tampilan saat menjalankan skenario. Gambar 5.48 menunjukkan tampilan tombol stop skenario. Gambar 5.49 menunjukkan tampilan setelah menekan tombol stop skenario.

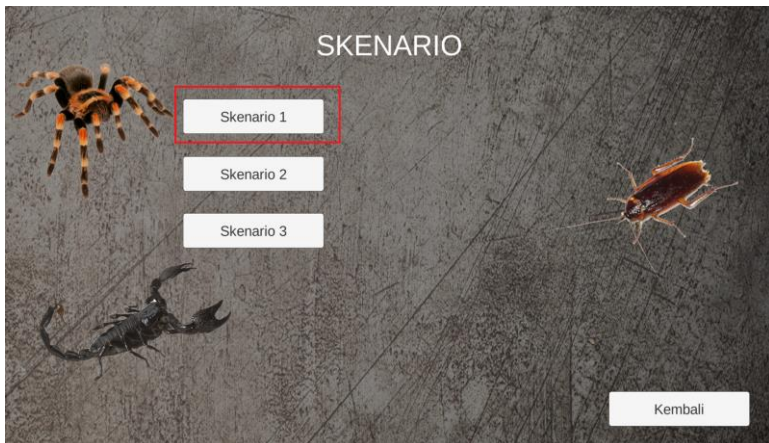
Tabel 5.13 Hasil uji coba menjalankan skenario

ID	UJ-UC-010
Nama	Uji Coba Menjalankan dan Menghentikan Skenario
Tujuan uji coba	Mengetahui apakah skenario dapat dijalankan dan dihentikan dengan benar
Kondisi awal	Sudah ada skenario yang ditambahkan Pengguna sudah masuk menu skenario
<i>Skenario 1</i>	<i>Pengguna menjalankan skenario sebagai terapis</i>
Masukan	Tekan pada tombol pilihan skenario yang ingin dijalankan
Keluaran yang diharapkan	Aplikasi akan menampilkan skenario yang dijalankan. Saat skenario dijalankan akan terdapat nama aksi yang sedang berjalan beserta durasi hitung mundur berjalannya aksi tersebut dan tombol stop skenario.
Hasil uji coba	Berhasil
Kondisi akhir	Aplikasi menampilkan skenario yang dijalankan. Saat skenario dijalankan terdapat nama aksi yang sedang berjalan beserta durasi hitung mundur berjalannya aksi tersebut dan tombol stop skenario.
<i>Skenario 2</i>	<i>Pengguna menjalankan skenario sebagai pasien</i>
Masukan	Tekan pada tombol pilihan skenario yang ingin dijalankan
Keluaran yang diharapkan	Aplikasi akan menampilkan skenario yang dijalankan. Saat skenario dijalankan hanya akan terdapat tombol stop skenario tanpa ada nama aksi dan durasi.
Hasil uji coba	Berhasil

ID	UJ-UC-010
Kondisi akhir	Aplikasi menampilkan skenario yang dijalankan. Saat skenario dijalankan hanya akan terdapat tombol stop skenario tanpa ada nama aksi dan durasi.
<i>Skenario 3</i>	<i>Pengguna menekan tombol stop skenario</i>
Masukan	Tekan pada tombol stop skenario
Keluaran yang diharapkan	Aplikasi akan menghentikan skenario yang berjalan. Tampilan aplikasi akan kembali ke menu skenario.
Hasil uji coba	Berhasil
Kondisi akhir	Aplikasi menghentikan skenario yang berjalan. Tampilan aplikasi kembali ke menu skenario.



Gambar 5.46 Tampilan pilihan skenario yang dapat dijalankan sebagai terapis



Gambar 5.47 Tampilan pilihan skenario yang dapat dijalankan sebagai pasien



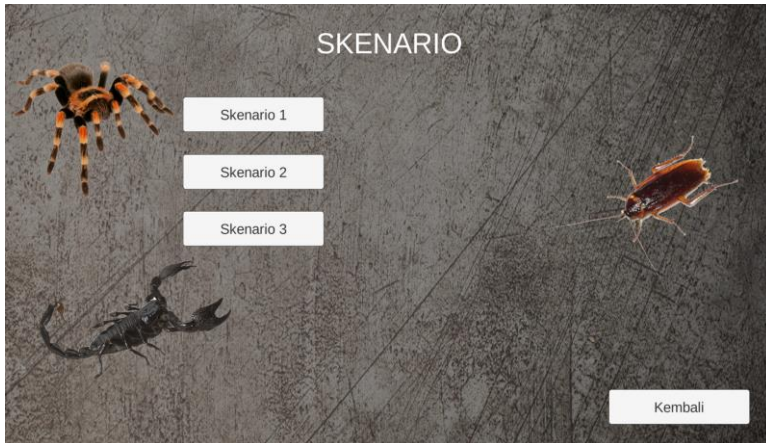
Gambar 5.48 Tampilan saat menjalankan skenario sebagai terapis



Gambar 5.49 Tampilan saat menjalankan skenario sebagai pasien



Gambar 5.50 Tampilan setelah menekan tombol stop skenario sebagai terapis



Gambar 5.51 Tampilan setelah menekan tombol stop skenario sebagai pasien

5.3 Pengujian Non-Fungsionalitas

Pengujian non-fungsionalitas dilakukan untuk mengetahui bagaimana hasil keluaran selain dari sisi non-fungsionalitas sistem terhadap skenario yang dipersiapkan. Berikut ini penjabaran skenario dan hasil uji coba non-fungsionalitas yang dilakukan terhadap perangkat lunak yang dibangun.

5.3.1 Skenario Uji Coba Non-Fungsionalitas

Subbab ini akan menjelaskan beberapa skenario uji coba perangkat lunak dengan mengadakan uji coba kepada beberapa orang yang menderita fobia serangga sebagai dasar tolak ukur keberhasilan. Pengujian non-fungsionalitas yang terdapat pada aplikasi dijabarkan sebagai berikut:

- Uji coba performa aplikasi
- Uji coba *immersive*
- Uji coba keefektifan aplikasi

Dari daftar pengujian yang telah disebutkan, akan dibuat beberapa skenario yang dilakukan pada setiap pengujian tersebut. Penjelasan mengenai cara dan hasil pengujian non-fungsionalitas perangkat lunak dibahas pada subbab 5.3.2.

5.3.2 Hasil Uji Coba

Subbab ini akan menjelaskan secara detail mengenai skenario yang dilakukan dan hasil yang didapat dari pengujian non-fungsionalitas perangkat lunak yang telah dibangun. Penjelasan akan ditampilkan berdasarkan hasil kuisioner yang telah diisi oleh partisipan saat proses terapi fobia serangga menggunakan aplikasi.

5.3.2.1 Daftar Penguji Perangkat Lunak

Pada subbab ini ditunjukkan daftar pengguna yang bertindak sebagai penguji coba aplikasi yang telah dibangun. Daftar nama penguji aplikasi ini dapat dilihat pada Tabel 5.14.

Tabel 5.14 Daftar nama penguji coba aplikasi

Responden	Pekerjaan
Eno Aji	Pengusaha
Donny Andrean	Musisi
Azzam Nasrul	Mahasiswa
Muhammad Yusron Ramadhan	Mahasiswa
Yusandi Rezki Fadli	Mahasiswa S2
Hermi	Mahasiswa
Muhammad Ali Anwar	Mahasiswa S2
Purna Hadi Swasono	Mahasiswa
Randytia	Mahasiswa
Oki	Event Organizer
Faris Muhammad Asmawi	Mahasiswa
Rahmad Cahyo Gumilar	Karyawan Swasta
Panji Andrean sundoro	Karyawan Swasta
Fajar Ainur Rofiq	Pengusaha

Ahmad Fauzi	Karyawan Swasta
Muhammad Khomarudin Bahar	Mahasiswa
Jordy	Mahasiswa
Eksels Van	Pelajar
Leadistia	Pelajar
Nisa	Mahasiswa

5.3.2.2 Hasil Uji Coba Performa Aplikasi

Uji coba performa aplikasi adalah ujicoba apakah aplikasi mampu memberikan rasa takut ke pengguna yang berupa penderita fobia serangga saat menjalani terapi fobia serangga menggunakan aplikasi. Penderita fobia serangga akan dipaparkan 8 macam presentasi serangga virtual dan 4 macam simulasi. Daftar presentasi dan simulasi yang dimaksud adalah sebagai berikut:

- 1 serangga kecil diam
- 1 serangga kecil berjalan
- Banyak serangga kecil diam
- Banyak serangga kecil berjalan
- 1 serangga besar diam
- 1 serangga besar berjalan
- Banyak serangga besar diam
- Banyak serangga besar berjalan
- Simulasi bunuh 1 serangga diam
- Simulasi bunuh 1 serangga berjalan
- Simulasi bunuh banyak serangga diam
- Simulasi bunuh banyak serangga berjalan

Saat terapi berlangsung, pada setiap kali selesai dengan 1 presentasi maupun simulasi, pengguna ditanyakan seberapa takut saat presentasi berlangsung dalam skala 1 sampai 10, dimana semakin besar nilai skala berarti semakin takut. Suasana saat terapi berlangsung ditunjukkan pada Gambar 5.52. Hasil uji coba akan dipaparkan pada Tabel 5.15.



Gambar 5.52 Uji coba terapi fobia serangga

Tabel 5.15 Rata-rata hasil uji coba performa aplikasi

Responden	Nilai Rata-rata
Eno Aji	5,42
Donny Andrean	6,42
Azzam Nasrul	7,42
Muhammad Yusron Ramadhan	7,08
Yusandi Rezki Fadli	6,92
Hermi	3,75
Muhammad Ali Anwar	8,25
Purna Hadi Swasono	7,08

Randytia	7,08
Oki	7,00
Faris Muhammad Asmawi	8,00
Rahmad Cahyo Gumilar	9,33
Panji Andrean sundoro	9,08
Fajar Ainur Rofiq	7,33
Ahmad Fauzi	4,50
Muhammad Khomarudin Bahar	6,50
Jordy	7,50
Eksels Van	8,42
Leadistia	8,25
Nisa	8,00
TOTAL	143,33
RATA-RATA	7.16

5.3.2.3 Hasil Uji Coba *Immersive*

Uji coba *immersive* adalah pengujian apakah pengguna bisa merasakan benar-benar berada di dunia nyata atau keadaan sebenarnya ketika menggunakan aplikasi yang telah dibangun. Hasil uji coba dipaparkan secara lengkap pada Tabel 5.16.

Tabel 5.16 Rata-rata hasil uji coba *immersive*

Responden	Nilai Rata-rata
Eno Aji	7,67
Donny Andrean	7,33
Azzam Nasrul	6,00
Muhammad Yusron Ramadhan	8,67
Yusandi Rezki Fadli	9,33
Hermi	7,00
Muhammad Ali Anwar	7,67
Purna Hadi Swasono	6,00
Randytia	6,67
Oki	7,67
Faris Muhammad Asmawi	8,67
Rahmad Cahyo Gumilar	8,67
Panji Andrean sundoro	9,00
Fajar Ainur Rofiq	6,67
Ahmad Fauzi	5,33
Muhammad Khomarudin Bahar	7,33
Jordy	6,33
Eksels Van	6,00
Leadistia	7,00
Nisa	7,67

TOTAL	146.67
RATA-RATA	7.33

5.3.2.4 Hasil Uji Coba Keefektifan Aplikasi

Uji coba keefektifan aplikasi adalah pengujian apakah aplikasi cukup efektif untuk digunakan sebagai pendukung terapi fobia serangga. Hasil uji coba dipaparkan secara lengkap pada Tabel 5.17.

Tabel 5.17 Rata-rata hasil uji coba keefektifan aplikasi

Responden	Nilai Kepuasan
Eno Aji	10
Donny Andrean	7
Azzam Nasrul	7
Muhammad Yusron Ramadhan	8
Yusandi Rezki Fadli	9
Hermi	7
Muhammad Ali Anwar	8
Purna Hadi Swasono	8
Randytia	8
Oki	8
Faris Muhammad Asmawi	9
Rahmad Cahyo Gumilar	9
Panji Andrean sundoro	7
Fajar Ainur Rofiq	7

Ahmad Fauzi	7
Muhammad Khomarudin Bahar	8
Jordy	5
Eksels Van	6
Leadistia	10
Nisa	10
TOTAL	158
RATA-RATA	7.9

5.4 Evaluasi

Subbab ini membahas mengenai evaluasi terhadap pengujian-pengujian yang telah dilakukan. Dalam hal ini, evaluasi menunjukkan data rekapitulasi dari hasil pengujian fungsionalitas dan pengujian non-fungsionalitas yang telah dilakukan sebelumnya.

5.4.1 Evaluasi Pengujian Fungsionalitas

Evaluasi pengujian fungsionalitas dilakukan dengan menampilkan data rekapitulasi perangkat lunak yang telah dipaparkan pada subbab 5.2. Dalam hal ini, rekapitulasi disusun dalam bentuk tabel yang dapat dilihat pada Tabel 5.18. Dari data yang terdapat pada tabel tersebut, diketahui bahwa aplikasi yang dibuat telah memenuhi kasus penggunaan yang telah ditentukan.

Tabel 5.18 Rekapitulasi hasil uji coba fungsionalitas

ID	Deskripsi		Hasil
UJ-UC-001	Uji Coba Menambah Serangga	Skenario 1	Berhasil
		Skenario 2	Berhasil
		Skenario 3	Berhasil
UJ-UC-002	Uji Coba Mengurangi Serangga	Skenario 1	Berhasil
UJ-UC-003	Uji Coba Menjalankan dan Menghentikan Serangga	Skenario 1	Berhasil
		Skenario 2	Berhasil
UJ-UC-004	Uji Coba Memperbesar Serangga	Skenario 1	Berhasil
UJ-UC-005	Uji Coba Memperkecil Serangga	Skenario 1	Berhasil
UJ-UC-006	Uji Coba Fitur Bunuh Serangga	Skenario 1	Berhasil
		Skenario 2	Berhasil
UJ-UC-007	Uji Coba Menambah Skenario	Skenario 1	Berhasil
		Skenario 2	Berhasil
		Skenario 3	Berhasil
		Skenario 4	Berhasil
		Skenario 5	Berhasil
		Skenario 6	Berhasil
UJ-UC-008	Uji Coba Mengubah Skenario	Skenario 1	Berhasil
		Skenario 2	Berhasil

ID	Deskripsi		Hasil
		Skenario 3	Berhasil
		Skenario 4	Berhasil
		Skenario 5	Berhasil
		Skenario 6	Berhasil
UJ-UC-009	Uji Coba Menghapus Skenario	Skenario 1	Berhasil
		Skenario 2	Berhasil
UJ-UC-010	Uji Coba Menjalankan dan Menghentikan Skenario	Skenario 1	Berhasil
		Skenario 2	Berhasil
		Skenario 3	Berhasil

5.4.2 Evaluasi Pengujian Non-Fungsionalitas

Evaluasi pengujian non-fungsionalitas dilakukan dengan menampilkan data rekapitulasi perangkat lunak yang telah dipaparkan pada subbab 5.2.3.7. Dalam hal ini, rekapitulasi disusun dalam bentuk tabel yang dapat dilihat pada Gambar 5.19. Dari data yang diketahui aplikasi telah memenuhi unsur *immersive* dan efektif sebagai pendukung terapi fobia serangga.

Tabel 5.19 Rekapitulasi hasil uji coba fungsionalitas

ID	Deskripsi	Rata-rata Nilai
UJ-UC-007	Uji Coba Performa Aplikasi	7.15
UJ-UC-008	Uji Coba <i>Immersive</i>	7.35
UJ-UC-009	Uji Coba Keefektifan Aplikasi	7.9

BAB VI

KESIMPULAN

Bab ini membahas mengenai kesimpulan yang dapat diambil dari tujuan pembuatan perangkat lunak dan hasil uji coba yang telah dilakukan sebagai jawaban dari rumusan masalah yang dikemukakan. Selain kesimpulan, terdapat pula saran yang ditujukan untuk pengembangan perangkat lunak lebih lanjut.

6.1. Kesimpulan

Dalam proses pengerjaan tugas akhir mulai dari tahap analisis, desain, implementasi, hingga pengujian didapatkan kesimpulan sebagai berikut:

1. Menurut hasil uji coba kotak hitam terhadap uji coba fungsionalitas, aplikasi ini sudah memenuhi ekspektasi.
2. Menurut hasil uji coba performa aplikasi, aplikasi ini sudah baik dalam menghadirkan rasa takut saat penderita fobia serangga menjalani terapi. Uji coba ini mendapatkan nilai rata-rata 7.16.
3. Menurut hasil uji coba *immersive*, aplikasi ini menciptakan keadaan yang seperti di dunia nyata. Uji coba ini mendapatkan nilai rata-rata 7.33.
4. Menurut hasil uji coba keefektifan aplikasi, aplikasi ini efektif digunakan sebagai pendukung terapi fobia serangga. Uji coba ini mendapatkan nilai rata-rata 7.9.

6.2. Saran

Berikut merupakan beberapa saran untuk pengembangan sistem di masa yang akan datang, berdasarkan pada hasil perancangan, implementasi dan uji coba yang telah dilakukan.

1. Panambahan pilihan serangga
2. Pengimplementasian dengan menggunakan kacamata VR

(Halaman ini sengaja dikosongkan)

DAFTAR PUSTAKA

- [1] I. Milosevic dan R. . E. McCabe, *Phobias: The Psychology of Irrational Fear: The Psychology of Irrational Fear*, ABC-CLIO, 2015.
- [2] infosehatkeluarga, “7 Cara Menghilangkan Phobia (Rasa Takut Berlebihan),” 2014. [Online]. Available: <http://www.infosehatkeluarga.com/7-cara-menghilangkan-phobia-rasa-takut-berlebihan/>.
- [3] M. Mekni dan A. Lemieux, “Augmented Reality: Applications, Challanges,” *Applied Computational Science*, pp. 205-214, 2014.
- [4] “SDK (Software Development Kit),” 21 6 2013. [Online]. Available: <http://gudanglinux.com/glossary/sdk-software-development-kit/>.
- [5] “Membuat game menggunakan software Unity 3D,” 3 2017. [Online]. Available: <http://www.tehnisikecil.com/2017/03/membuat-game-menggunakan-software-unity.html>.
- [6] “Unity - Engine Features,” [Online]. Available: <https://unity3d.com/unity/engine-features>.
- [7] A. W. Putra, “Vuforia – SDK Canggih Untuk Wujudkan Aplikasi dan Game Dengan Teknologi Augmented Reality,” 30 4 2015. [Online]. Available: <https://teknojurnal.com/vuforia/>.

- [8] I. Setiwan, *Perancangan Software Embedded System Berbasis FSM*, 2006.
- [9] J. B. Lopez, S. Quero, C. Botella dan A. G. Palacios, "An Augmented Reality System Validation for The Treatment of Cockroach Phobia," *Cyberpsychology, Behaviour, And Social Networking*, 2010.
- [10] M. Wrzesien, M. Alcaniz, C. Botella, J. M. Burkhardt, J. B. Lopez, M. Ortega dan B. Brotons, "The Therapeutic Lamp: Treating Small-Animal Phobias," *IEEE Computer Society*, 2013.
- [11] Budiyanto, "Pemanfaatan Augmented Reality Sebagai Media Terapi Penderita Fobia Terhadap Hewan," *Jurnal Ilmiah Komputer dan Informatika*, 2015.

LAMPIRAN

```

1  using UnityEngine;
2  using System.Collections;
3  using UnityEngine.UI;
4
5  public class appmanager : MonoBehaviour {
6      int sizedefault, countInsect, rotasirand, flag,
7      maxInsect, moveinsectstat;
8      float scaleupvalue, scaledownvalue,
9      maxsizeinsect, minsizeinsect,
10     currentaddsizeinsect, insectDefaultSize,
11     insectSize;
12     public GameObject [] insect;
13     public Animator [] spideranim, cockroachanim,
14     scorpioanim, tarantulaanim, cricketanim;
15     Vector3 spawnrot, spawnsize;
16     int settingButtonStatus, insectSizeShowned,
17     cekSeranggaJalan, cekBunuhSerangga;
18     public GameObject settingContent;
19     public GameObject fences1, fences2, fences3,
20     fences4;
21     public GameObject
22     jumlahSeranggaValue, moveInsectLabel,
23     staticInsectLabel, ukuranSeranggaValue,
24     bunuhSeranggaValue,
25     showmoreinsectContent, settingButton,
26     jumlahSeranggaValue2, seranggajalanButtonTxt,
27     bunuhSeranggaButtonTxt;
28
29     void Start () {
30         maxInsect = 10;
31         spawnsize = new Vector3 (70, 1000, 70);
32         scaleupvalue = 0.3f;
33         scaledownvalue = 0.3f;
34         maxsizeinsect = 2.7f;
35         minsizeinsect = 1.7f;
36         insectSize = 1.7f;
37         insectSizeShowned = 1;
38         if (flag == 0) {
39             for (int k = 0; k < 10; k++) {
40                 insect[k].gameObject.SetActive (false);
41             }
42             flag = 1;
43         }
44         if (flag == 1) {
45             flag = -1;
46             disableVB ();
47         }
48     }
49     public void showMoreInsect(int insectChoser){

```

```

50         if (countInsect < maxInsect) {
51             if (!insect[0].activeSelf) {
52                 insect[0].gameObject.SetActive (true);
53                 if (insectChoser != -1) {
54                     insect[0].transform.GetChild
55                         (insectChoser).gameObject.SetActive
56                         (true);
57                 }
58                 spawnrot = new Vector3 (0, Random.Range
59                     (0, 360), 0);
60                 insect[0].gameObject.transform.Rotate
61                     (spawnrot);
62             } else {
63                 for (int f = 1; f < 10; f++) {
64                     if (!insect [f].activeSelf) {
65                         while (!insect [f].activeSelf) {
66                             Vector3 spawnpos = new Vector3
67                                 (Random.Range (-8, 8), 0,
68                                 Random.Range (-8, 8));
69                             if (Physics.OverlapBox (spawnpos *
70                                 100, spawnsize).Length == 0) {
71                                 insect [f].gameObject.SetActive
72                                 (true);
73                                 if (insectChoser != -1) {
74                                     insect [f].transform.GetChild
75                                         (insectChoser).gameObject.
76                                         SetActive(true);
77                                 }
78                                 insect[f].gameObject.transform.
79                                 localPosition = spawnpos / 10;
80                                 spawnrot = new Vector3 (0,
81                                 Random.Range (0, 360), 0);
82                                 insect [f].gameObject.transform.
83                                 Rotate (spawnrot);
84                             }
85                         }
86                         break;
87                     }
88                 }
89             }
90             countInsect++;
91             jumlahSeranggaValue.GetComponent<Text>
92             ().text = countInsect.ToString() + "/10";
93             jumlahSeranggaValue2.GetComponent<Text>
94             ().text = countInsect.ToString() + "/10";
95         }
96         if (moveinsectstat == 1) {
97             moveInsect ();
98         }

```

```

99         if (moveinsectstat == 0) {
100             staticInsect ();
101         }
102     }
103     void showlessinsect(){
104         if (countInsect > 0) {
105             insect[countInsect-1].gameObject.SetActive
106             (false);
107             for (int l = 1; l < 6; l++) {
108                 insect[countInsect-1].transform.
109                 GetChild(l).gameObject.SetActive (false)
110             }
111             countInsect--;
112             jumlahSeranggaValue.GetComponent<Text>
113             ().text = countInsect.ToString () + "/10";
114             jumlahSeranggaValue2.GetComponent<Text>
115             ().text = countInsect.ToString() + "/10";
116         }
117     }
118     void moveInsect(){
119         staticInsectLabel.gameObject.SetActive (false);
120         moveInsectLabel.gameObject.SetActive (true);
121         moveinsectstat = 1;
122         for (int j = 0; j < 10; j++) {
123             (insect[j].GetComponent ("moveInsect") as
124             MonoBehaviour).enabled = true;
125             spideranim[j].SetInteger ("insectstate", 1);
126             cockroachanim[j].SetInteger ("insectstate",
127             1);
128             scorpioanim[j].SetInteger ("insectstate", 1);
129             tarantulaanim[j].SetInteger ("insectstate",
130             1);
131             cricketanim[j].SetInteger ("insectstate", 1);
132         }
133     }
134     void staticInsect(){
135         staticInsectLabel.gameObject.SetActive (true);
136         moveInsectLabel.gameObject.SetActive (false);
137         moveinsectstat = 0;
138         for (int j = 0; j < 10; j++) {
139             (insect[j].GetComponent ("moveInsect") as
140             MonoBehaviour).enabled = false;
141             spideranim[j].SetInteger ("insectstate", 0);
142             cockroachanim[j].SetInteger ("insectstate",
143             0);
144             scorpioanim[j].SetInteger ("insectstate", 0);
145             tarantulaanim[j].SetInteger ("insectstate",
146             0);
147             cricketanim[j].SetInteger ("insectstate", 0);

```

```

148     }
149 }
150 void biggerInsect(){
151     if (insectSize < maxsizeinsect) {
152         insectSize += scaleupvalue;
153         insectSizeShown++;
154         ukuranSeranggaValue.GetComponent<Text>
155         ().text = insectSizeShown.ToString () +
156         "/5";
157         for (int p = 0; p < 9; p++) {
158             insect[p].transform.localScale = new
159             Vector3 (insectSize,
160             insectSize,insectSize);
161         }
162     }
163 }
164 void smallerInsect(){
165     if (insectSize > minsizeinsect) {
166         insectSize -= scaleupvalue;
167         insectSizeShown--;
168         ukuranSeranggaValue.GetComponent<Text>
169         ().text = insectSizeShown.ToString () +
170         "/5";
171         for (int p = 0; p < 9; p++) {
172             insect[p].transform.localScale = new
173             Vector3 (insectSize,
174             insectSize,insectSize);
175         }
176     }
177 }
178 public void respawn(){
179     if (insect[0].gameObject.GetComponent
180     <moveInsect> ().dead == 1) {
181         insect[0].gameObject.GetComponent<moveInsect>
182         ().dead = 0;
183         while (!insect[0].activeSelf) {
184             Vector3 spawnpos = new Vector3
185             (Random.Range (-8, 8), 0, Random.Range (-8,
186             8));
187             if (Physics.OverlapBox (spawnpos * 100,
188             spawnsize).Length == 0) {
189                 insect[0].gameObject.SetActive (true);
190                 insect[0].gameObject.transform.
191                 localPosition = spawnpos / 10;
192                 spawnrot = new Vector3 (0, Random.Range
193                 (0, 360), 0);
194                 insect[0].gameObject.transform.Rotate
195                 (spawnrot);
196             }

```

```

197     }
198     if (moveinsectstat == 1) {
199         moveInsect ();
200     }
201     if (moveinsectstat == 0) {
202         staticInsect ();
203     }
204 } else {
205     for (int r = 1; r < 10; r++) {
206         if (insect[r].gameObject.GetComponent
207             <moveInsect> ().dead == 1) {
208             insect[r].gameObject.GetComponent
209                 <moveInsect> ().dead = 0;
210             countInsect--;
211             showMoreInsect (-1);
212             break;
213         }
214     }
215 }
216 }
217 void resetButton(){
218     Vector3 spawnpos = new Vector3 (0, 0, 0);
219     insect[0].gameObject.transform.localPosition =
220     spawnpos;
221 }
222 public void resetAll(){
223     for (int j = 0; j <= 10; j++) {
224         showlessinsect ();
225         if (j <= 5) {
226             smallerInsect ();
227         }
228     }
229     cekSeranggaJalan = 0;
230     cekBunuhSerangga = 0;
231     staticInsect ();
232     disableVB ();
233     resetButton ();
234 }
235 public void enableVB(){
236     bunuhSeranggaValue.GetComponent<Text>().text =
237     "ON";
238     for (int g = 0; g < 10; g++) {
239         (insect[g].transform.GetChild(0).gameObject.
240             GetComponent("VirtualButtonBehaviour") as
241             MonoBehaviour).enabled = true;
242     }
243 }
244 public void disableVB(){
245     bunuhSeranggaValue.GetComponent<Text>().text =

```



```

246 "OFF";
247 for (int g = 0; g < 10; g++) {
248     (insect[g].transform.GetChild(0).gameObject.
249     GetComponent("VirtualButtonBehaviour") as
250     MonoBehaviour).enabled = false;
251 }
252 }
253 //-----
254 -----ButtonAction-----
255 -----//
256 public void settingButtonAction(){
257     settingContent.SetActive (true);
258     settingButton.SetActive (false);
259 }
260 public void keluarButtonAction(){
261     settingContent.SetActive (false);
262     settingButton.SetActive (true);
263 }
264 public void showMoreInsectButtonAction(){
265     showmoreinsectContent.SetActive (true);
266     settingContent.SetActive (false);
267     settingButton.SetActive (false);
268 }
269 public void showLessInsectButtonAction(){
270     showlessinsect ();
271 }
272 public void moveInsectButtonAction(){
273     if (cekSeranggaJalan == 0) {
274         moveInsect ();
275         cekSeranggaJalan = 1;
276         seranggajalanButtonTxt.GetComponent<Text>
277         ().text = "Serangga Diam";
278     } else {
279         staticInsect ();
280         cekSeranggaJalan = 0;
281         seranggajalanButtonTxt.GetComponent<Text>
282         ().text = "Serangga Jalan";
283     }
284 }
285 public void staticInsectButtonAction(){
286     staticInsect ();
287 }
288 public void biggerInsectButtonAction(){
289     biggerInsect ();
290 }
291 public void smallerInsectButtonAction(){
292     smallerInsect ();
293 }
294 public void enableVBButtonAction(){

```

295	if (cekBunuhSerangga == 0) {
296	enableVB ();
297	cekBunuhSerangga = 1;
298	bunuhSeranggaButtonTxt.GetComponent<Text>
299	().text = "Bunuh Serangga: OFF";
300	} else {
301	disableVB();
302	cekBunuhSerangga = 0;
303	bunuhSeranggaButtonTxt.GetComponent<Text>
304	().text = "Bunuh Serangga: ON";
305	}
306	}
307	public void disableVBButtonAction(){
308	disableVB ();
309	}
310	public void insectChoserButtonAction(int a){
311	showMoreInsect (a);
312	}
313	public void backButtonAction(){
314	showmoreinsectContent.SetActive (false);
315	settingContent.SetActive (true);
316	}
317	public void resetButtonAction(){
318	resetButton ();
319	}
320	}

Gambar A.1 Script appmanager

```

1  using UnityEngine;
2  using System.Collections;
3
4  public class moveInsect : MonoBehaviour {
5      public float speed;
6      public Vector3 originalpos;
7      public int flag, dead;
8      private Rigidbody rbody;
9      void Start () {
10         rbody = GetComponent<Rigidbody> ();
11     }
12     void Update () {
13         Vector3 pos = gameObject.transform.localPosition;
14         transform.Translate (Vector3.forward * speed *
15             Time.deltaTime);
16         if (gameObject.transform.localPosition.x < -1 ||
17             gameObject.transform.localPosition.x > 1 ||
18             gameObject.transform.localPosition.z < -1 ||
19             gameObject.transform.localPosition.z > 1) {
20             Vector3 rotasi = new Vector3 (0, 90, 0);
21             gameObject.transform.Rotate (rotasi);
22         }
23     }
24     void OnTriggerEnter(Collider other){
25         Vector3 rotasi = new Vector3 (0, 210, 0);
26         gameObject.transform.Rotate (rotasi);
27     }
28 }

```

Gambar A.2 Script *moveInsect*

```

1 using UnityEngine;
2 using System.Collections;
3
4 namespace Vuforia{
5 public class VBEventHandler : MonoBehaviour,
6 IVirtualButtonEventHandler {
7     public GameObject insect;
8     string insectname;
9     int insectcount, flag;
10    VirtualButtonBehaviour[] vbs;
11    void Start(){
12        vbs = GetComponentsInChildren
13            <VirtualButtonBehaviour>();
14        for (int i = 0; i < vbs.Length; ++i) {
15            vbs[i].RegisterEventHandler(this);
16        }
17        insect = transform.FindChild("appManager").
18            gameObject;
19    }
20    IEnumerator deadAction(int insectnum) {
21        yield return new WaitForSeconds(1);
22        insect.transform.GetChild(insectnum).gameObject.
23            GetComponent<moveInsect> ().dead = 1;
24        insect.transform.GetChild(insectnum).gameObject.
25            SetActive(false);
26        insect.GetComponent<appmanager> ().respawn ();
27    }
28    public void OnButtonPressed
29    (VirtualButtonAbstractBehaviour vb){
30        if (vb!= null) {
31            switch (vb.VirtualButtonName) {
32                case "deadinsect1":
33                    insect.transform.GetChild
34                        (0).transform.GetChild(1).
35                        GetComponent<Animator> ().SetInteger
36                            ("insectstate", 2);
37                    insect.transform.GetChild
38                        (0).transform.GetChild(2).GetComponent
39                            <Animator> ().SetInteger ("insectstate",
40                                2);
41                    insect.transform.GetChild
42                        (0).transform.GetChild(3).GetComponent
43                            <Animator> ().SetInteger ("insectstate",
44                                2);
45                    insect.transform.GetChild
46                        (0).transform.GetChild(4).GetComponent
47                            <Animator> ().SetInteger ("insectstate",
48                                2);
49                    insect.transform.GetChild

```

50	(0).transform.GetChild(5).GetComponent
51	<Animator> ().SetInteger ("insectstate",
52	2);
53	(insect.transform.GetChild(0).GetComponent
54	("moveInsect") as MonoBehaviour).
55	enabled = false;
56	break;
57	case "deadinsect2":
58	insect.transform.GetChild
59	(1).transform.GetChild(1).
60	GetComponent<Animator> ().SetInteger
61	("insectstate", 2);
62	insect.transform.GetChild
63	(1).transform.GetChild(2).GetComponent
64	<Animator> ().SetInteger ("insectstate",
65	2);
66	insect.transform.GetChild
67	(1).transform.GetChild(3).GetComponent
68	<Animator> ().SetInteger ("insectstate",
69	2);
70	insect.transform.GetChild
71	(1).transform.GetChild(4).GetComponent
72	<Animator> ().SetInteger ("insectstate",
73	2);
74	insect.transform.GetChild
75	(1).transform.GetChild(5).GetComponent
76	<Animator> ().SetInteger ("insectstate",
77	2);
78	(insect.transform.GetChild(1).GetComponent
79	("moveInsect") as MonoBehaviour).
80	enabled = false;
81	break;
82	case "deadinsect3":
83	insect.transform.GetChild
84	(2).transform.GetChild(1).
85	GetComponent<Animator> ().SetInteger
86	("insectstate", 2);
87	insect.transform.GetChild
88	(2).transform.GetChild(2).GetComponent
89	<Animator> ().SetInteger ("insectstate",
90	2);
91	insect.transform.GetChild
92	(2).transform.GetChild(3).GetComponent
93	<Animator> ().SetInteger ("insectstate",
94	2);
95	insect.transform.GetChild
96	(2).transform.GetChild(4).GetComponent
97	<Animator> ().SetInteger ("insectstate",
98	2);

99	insect.transform.GetChild
100	(2).transform.GetChild(5).GetComponent
101	<Animator> ().SetInteger ("insectstate",
102	2);
103	(insect.transform.GetChild(2).GetComponent
104	("moveInsect") as MonoBehaviour).
105	enabled = false;
106	break;
107	case "deadinsect4":
108	insect.transform.GetChild
109	(3).transform.GetChild(1).
110	GetComponent<Animator> ().SetInteger
111	("insectstate", 2);
112	insect.transform.GetChild
113	(3).transform.GetChild(2).GetComponent
114	<Animator> ().SetInteger ("insectstate",
115	2);
116	insect.transform.GetChild
117	(3).transform.GetChild(3).GetComponent
118	<Animator> ().SetInteger ("insectstate",
119	2);
120	insect.transform.GetChild
121	(3).transform.GetChild(4).GetComponent
122	<Animator> ().SetInteger ("insectstate",
123	2);
124	insect.transform.GetChild
125	(3).transform.GetChild(5).GetComponent
126	<Animator> ().SetInteger ("insectstate",
127	2);
128	(insect.transform.GetChild(3).GetComponent
129	("moveInsect") as MonoBehaviour).
130	enabled = false;
131	break;
132	case "deadinsect5":
133	insect.transform.GetChild
134	(4).transform.GetChild(1).
135	GetComponent<Animator> ().SetInteger
136	("insectstate", 2);
137	insect.transform.GetChild
138	(4).transform.GetChild(2).GetComponent
139	<Animator> ().SetInteger ("insectstate",
140	2);
141	insect.transform.GetChild
142	(4).transform.GetChild(3).GetComponent
143	<Animator> ().SetInteger ("insectstate",
144	2);
145	insect.transform.GetChild
146	(4).transform.GetChild(4).GetComponent
147	<Animator> ().SetInteger ("insectstate",

```

148         2);
149         insect.transform.GetChild
150         (4).transform.GetChild(5).GetComponent
151         <Animator> ().SetInteger ("insectstate",
152         2);
153         (insect.transform.GetChild(4).GetComponent
154         ("moveInsect") as MonoBehaviour).
155         enabled = false;
156         break;
157     case "deadinsect6":
158         insect.transform.GetChild
159         (5).transform.GetChild(1).
160         GetComponent<Animator> ().SetInteger
161         ("insectstate", 2);
162         insect.transform.GetChild
163         (5).transform.GetChild(2).GetComponent
164         <Animator> ().SetInteger ("insectstate",
165         2);
166         insect.transform.GetChild
167         (5).transform.GetChild(3).GetComponent
168         <Animator> ().SetInteger ("insectstate",
169         2);
170         insect.transform.GetChild
171         (5).transform.GetChild(4).GetComponent
172         <Animator> ().SetInteger ("insectstate",
173         2);
174         insect.transform.GetChild
175         (5).transform.GetChild(5).GetComponent
176         <Animator> ().SetInteger ("insectstate",
177         2);
178         (insect.transform.GetChild(5).GetComponent
179         ("moveInsect") as MonoBehaviour).
180         enabled = false;
181         break;
182     case "deadinsect7":
183         insect.transform.GetChild
184         (6).transform.GetChild(1).
185         GetComponent<Animator> ().SetInteger
186         ("insectstate", 2);
187         insect.transform.GetChild
188         (6).transform.GetChild(2).GetComponent
189         <Animator> ().SetInteger ("insectstate",
190         2);
191         insect.transform.GetChild
192         (6).transform.GetChild(3).GetComponent
193         <Animator> ().SetInteger ("insectstate",
194         2);
195         insect.transform.GetChild
196         (6).transform.GetChild(4).GetComponent

```

197	<Animator> ().SetInteger ("insectstate",
198	2);
199	insect.transform.GetChild
200	(6).transform.GetChild(5).GetComponent
201	<Animator> ().SetInteger ("insectstate",
202	2);
203	(insect.transform.GetChild(6).GetComponent
204	("moveInsect") as MonoBehaviour).
205	enabled = false;
206	break;
207	case "deadinsect8":
208	insect.transform.GetChild
209	(7).transform.GetChild(1).
210	GetComponent<Animator> ().SetInteger
211	("insectstate", 2);
212	insect.transform.GetChild
213	(7).transform.GetChild(2).GetComponent
214	<Animator> ().SetInteger ("insectstate",
215	2);
216	insect.transform.GetChild
217	(7).transform.GetChild(3).GetComponent
218	<Animator> ().SetInteger ("insectstate",
219	2);
220	insect.transform.GetChild
221	(7).transform.GetChild(4).GetComponent
222	<Animator> ().SetInteger ("insectstate",
223	2);
224	insect.transform.GetChild
225	(7).transform.GetChild(5).GetComponent
226	<Animator> ().SetInteger ("insectstate",
227	2);
228	(insect.transform.GetChild(7).GetComponent
229	("moveInsect") as MonoBehaviour).
230	enabled = false;
231	break;
232	case "deadinsect9":
233	insect.transform.GetChild
234	(8).transform.GetChild(1).
235	GetComponent<Animator> ().SetInteger
236	("insectstate", 2);
237	insect.transform.GetChild
238	(8).transform.GetChild(2).GetComponent
239	<Animator> ().SetInteger ("insectstate",
240	2);
241	insect.transform.GetChild
242	(8).transform.GetChild(3).GetComponent
243	<Animator> ().SetInteger ("insectstate",
244	2);
245	insect.transform.GetChild


```

246         (8).transform.GetChild(4).GetComponent
247         <Animator> ().SetInteger ("insectstate",
248         2);
249         insect.transform.GetChild
250         (8).transform.GetChild(5).GetComponent
251         <Animator> ().SetInteger ("insectstate",
252         2);
253         (insect.transform.GetChild(8).GetComponent
254         ("moveInsect") as MonoBehaviour).
255         enabled = false;
256         break;
257     case "deadinsect10":
258         insect.transform.GetChild
259         (9).transform.GetChild(1).
260         GetComponent<Animator> ().SetInteger
261         ("insectstate", 2);
262         insect.transform.GetChild
263         (9).transform.GetChild(2).GetComponent
264         <Animator> ().SetInteger ("insectstate",
265         2);
266         insect.transform.GetChild
267         (9).transform.GetChild(3).GetComponent
268         <Animator> ().SetInteger ("insectstate",
269         2);
270         insect.transform.GetChild
271         (9).transform.GetChild(4).GetComponent
272         <Animator> ().SetInteger ("insectstate",
273         2);
274         insect.transform.GetChild
275         (9).transform.GetChild(5).GetComponent
276         <Animator> ().SetInteger ("insectstate",
277         2);
278         (insect.transform.GetChild(9).GetComponent
279         ("moveInsect") as MonoBehaviour).
280         enabled = false;
281         break;
282     }
283 }
284 }
285 public void OnButtonReleased
286 (VirtualButtonAbstractBehaviour vb){
287     if (vb!=null) {
288         switch (vb.VirtualButtonName) {
289             case "deadinsect1":
290                 StartCoroutine (deadAction (0));
291                 break;
292             case "deadinsect2":
293                 StartCoroutine (deadAction (1));
294                 break;

```

295	case "deadinsect3":
296	StartCoroutine (deadAction (2));
297	break;
298	case "deadinsect4":
299	StartCoroutine (deadAction (3));
300	break;
301	case "deadinsect5":
302	StartCoroutine (deadAction (4));
303	break;
304	case "deadinsect6":
305	StartCoroutine (deadAction (5));
306	break;
307	case "deadinsect7":
308	StartCoroutine (deadAction (6));
309	break;
310	case "deadinsect8":
311	StartCoroutine (deadAction (7));
312	break;
313	case "deadinsect9":
314	StartCoroutine (deadAction (8));
315	break;
316	case "deadinsect10":
317	StartCoroutine (deadAction (9));
318	break;
319	}
320	}
321	}
322	}
323	}

Gambar A.3 Script VBEventHandler

```

1  using System.Collections;
2  using System.Collections.Generic;
3  using UnityEngine;
4  using UnityEngine.UI;
5  public class scenarioManager : MonoBehaviour {
6      public GameObject scenarioButton, addScenarioButton,
7          stopSkenario, settingButton, scenarioAction,
8          scenarioTimer, dataBelumLengkap, jumlahLabel,
9          dataWrong, mainMenuContent, ubahScenarioButton,
10         hapusScenarioButton, backToMainMenuButton,
11         creditContent, bgImage, terapisContent,
12         menuContent, jmlSeranggaValue, scaleValue,
13         insectWarning, scaleWarning, tutorialContent;
14         public GameObject actionDropdown,
15             jumlahSeranggaDropdown, insectSizeDropdown,
16             insectSizeDropdown2, durationAddScenario,
17             addInsectDropdown, nextScenarioDropdown,
18             actionLabelAddScenario, addScenarioLabel,
19             kurangiSeranggaDropdown, nextActionButton,
20             prevActionButton;
21         public GameObject scenarioContent,
22             addScenarioContent, settingContent;
23         public GameObject [] tambahSeranggaDropdown,
24             skenario, scenarioview, scenarioedit,
25             scenariodelete;
26         public GameObject ubahSkenarioContent,
27             ubahSkenarioContentEditor, hapusSkenarioContent;
28         string[] eachscenario, action, value, duration,
29             scenarioTemp;
30         string scenario, allscenario, durationValue,
31             saveScenario, allAction;
32         int count, actionValue, insertValue, insectIndex,
33             actionCount, actionCountPrint, scenarioCount,
34             currentAction, actionLength;
35         int [] insectCount, addInsect, scaleInsect;
36         int insectSize, idx, flagedit, indexedit,
37             insertScenario, jumlahSkenario, isPasien,
38             totalInsect, totalScale, showedAction,
39             somethingWrong;
40         void Start () {
41             if (PlayerPrefs.GetInt ("firstTime")==0) {
42                 PlayerPrefs.SetString ("scenario0", "0a0v5d2a-
43                     v5d3a-v5d");
44                 PlayerPrefs.SetInt ("firstTime", 1);
45             }
46             actionDropdown.GetComponent<Dropdown>
47                 ().onValueChanged.AddListener (delegate {
48                 changeAction ();});
49             jumlahSeranggaDropdown.GetComponent<Dropdown>().

```

```

50         onValueChanged.AddListener(delegate {
51             showInsectChoice ();});
52         scenarioTemp = new string[25];
53         insectCount = new int[25];
54         addInsect = new int[25];
55         scaleInsect = new int[25];
56     }
57     public void changeAction(){
58         if (actionDropdown.activeSelf) {
59             jumlahLabel.SetActive (true);
60             insectSizeDropdown.SetActive (false);
61             insectSizeDropdown2.SetActive (false);
62             kurangiSeranggaDropdown.SetActive (false);
63             jumlahSeranggaDropdown.SetActive (false);
64             if (actionDropdown.GetComponent<Dropdown>
65                 ().value == 0) {
66                 jumlahSeranggaDropdown.SetActive (true);
67                 defaultformaddscenario ();
68             }
69             else if (actionDropdown.GetComponent<Dropdown>
70                 ().value == 1) {
71                 kurangiSeranggaDropdown.SetActive (true);
72                 deactivateAllChildAddScenario ();
73             }
74             else if (actionDropdown.GetComponent<Dropdown>
75                 ().value == 4) {
76                 insectSizeDropdown.SetActive (true);
77                 deactivateAllChildAddScenario ();
78             }
79             else if (actionDropdown.GetComponent<Dropdown>
80                 ().value == 5) {
81                 insectSizeDropdown2.SetActive (true);
82                 deactivateAllChildAddScenario ();
83             }
84             else {
85                 jumlahLabel.SetActive (false);
86                 deactivateAllChildAddScenario ();
87             }
88         }
89     }
90     public void deactivateAllChildAddScenario(){
91         for (int y = 0; y <= 9; y++) {
92             addInsectDropdown.transform.GetChild
93                 (y).gameObject.SetActive (false);
94         }
95     }
96     public void showInsectChoice() {
97         if (jumlahSeranggaDropdown.activeSelf) {
98             for (int x = 0; x <= jumlahSeranggaDropdown.

```

```

99         GetComponent<Dropdown> ().value; x++) {
100             addInsectDropdown.transform.GetChild
101                 (x).gameObject.SetActive (true);
102         }
103         for (int y = 9; y > jumlahSeranggaDropdown.
104             GetComponent<Dropdown> ().value; y--) {
105             addInsectDropdown.transform.GetChild (y).
106                 gameObject.SetActive (false);
107         }
108     }
109 }
110 public void getEachScenario(string allscenario){
111     eachscenario = allscenario.Split ('d');
112     int length = eachscenario.Length;
113     actionLength = eachscenario.Length;
114     action = new string[length];
115     value = new string[length];
116     duration = new string[length];
117     count = 0;
118     while (length>1) {
119         action[count] = eachscenario[count].
120             Substring(0,1);
121         value[count] = eachscenario[count].
122             Substring(2,1);
123         duration[count] = eachscenario[count].
124             Substring(4);
125         count++;
126         length--;
127     }
128 }
129 IEnumerator runAction() {
130     idx = 0;
131     stopSkenario.SetActive (true);
132     settingButton.SetActive (false);
133     if (isPasien == 0) {
134         scenarioAction.SetActive (true);
135         scenarioTimer.SetActive (true);
136     }
137     while(idx<count){
138         int durationnn = int.Parse(duration[idx]);
139         if (action [idx] == "0") {//jika aksi =
140             menambah serangan
141             scenarioAction.GetComponent<Text> ().text =
142                 "Menambah Serangga";
143             int val = int.Parse (value [idx]) + 1;
144             this.gameObject.GetComponent<appmanager>
145                 ().showMoreInsect (val);
146             } else if (action [idx] == "1") {//jika aksi
147                 = mengurangi serangan

```

```

148         scenarioAction.GetComponent<Text> ().text =
149         "Mengurangi Serangga";
150         int val = int.Parse (value [idx]);
151         for (int n = 0; n < val; n++) {
152             this.gameObject.GetComponent<appmanager>
153             ().showLessInsectButtonAction ();
154         }
155     } else if (action [idx] == "2") {//jika aksi =
156     serangga berjalan
157         scenarioAction.GetComponent<Text> ().text =
158         "Serangga Berjalan";
159         this.gameObject.GetComponent<appmanager>
160         ().moveInsectButtonAction ();
161     } else if (action [idx] == "3") {//jika aksi
162     = serangga diam
163         scenarioAction.GetComponent<Text> ().text =
164         "Serangga Diam";
165         this.gameObject.GetComponent<appmanager>
166         ().staticInsectButtonAction ();
167     } else if (action [idx] == "4") {//jika aksi =
168     perbesar serangga
169         scenarioAction.GetComponent<Text> ().text =
170         "Perbesar Serangga";
171         int val = int.Parse (value [idx]);
172         for (int n = 0; n < val; n++) {
173             this.gameObject.GetComponent<appmanager>
174             ().biggerInsectButtonAction ();
175         }
176     } else if (action [idx] == "5") {//jika aksi =
177     perkecil serangga
178         scenarioAction.GetComponent<Text> ().text =
179         "Perkecil Serangga";
180         int val = int.Parse (value [idx]);
181         for (int n = 0; n < val; n++) {
182             this.gameObject.GetComponent<appmanager>
183             ().smallerInsectButtonAction ();
184         }
185     } else if (action [idx] == "6") {//jika aksi =
186     bunuh serangga:on
187         scenarioAction.GetComponent<Text> ().text =
188         "Bunuh Serangga: ON";
189         this.gameObject.GetComponent<appmanager>
190         ().enableVB ();
191     } else if (action [idx] == "7") {//jika aksi =
192     bunuh serangga:off
193         scenarioAction.GetComponent<Text> ().text =
194         "Bunuh Serangga: OFF";
195         this.gameObject.GetComponent<appmanager>
196         ().disableVB ();

```

```

197         } else if (action [idx] == "8") { //jika aksi =
198         reset posisi serangga pertama
199         scenarioAction.GetComponent<Text> ().text =
200         "Reset Posisi Serangga 1";
201         this.gameObject.GetComponent<appmanager>
202         ().resetButtonAction ();
203     }
204     idx++;
205     while (durationnn != 0) {
206         scenarioTimer.GetComponent<Text> ().text =
207         durationnn.ToString();
208         yield return new WaitForSeconds (1);
209         durationnn--;
210     }
211 }
212 stopScenario ();
213 this.gameObject.GetComponent<appmanager>
214 ().resetAll();
215 if (isPasien == 1) {
216     scenarioButtonAction ();
217 }
218 }
219 public void runScenario(){
220     StopAllCoroutines ();
221     scenarioContent.SetActive (false);
222     bgImage.SetActive (false);
223     this.gameObject.GetComponent<appmanager>
224     ().resetAll();
225     StartCoroutine (runAction ());
226 }
227 public void stopScenario(){
228     StopAllCoroutines ();
229     this.gameObject.GetComponent<appmanager>
230     ().resetAll();
231     settingButton.SetActive (true);
232     stopSkenario.SetActive (false);
233     scenarioAction.SetActive (false);
234     scenarioTimer.SetActive (false);
235     scenarioButtonAction ();
236 }
237 public void insertScenarioTemp(string inValue){
238     scenarioTemp [currentAction] +=
239     actionValue.ToString () + "a" +
240     inValue + "v" + durationValue + "d";
241 }
242 public int addInsectTotal(int vals){
243     totalInsect = 0;
244     int tooMuch = 0;
245     for (int y = 0; y < vals; y++) {

```

```

246         totalInsect += addInsect [y];
247         if (totalInsect > 10 || totalInsect < 0) {
248             tooMuch = 1;
249             break;
250         }
251     }
252     if (tooMuch == 0) {
253         return totalInsect;
254     } else {
255         return -1;
256     }
257 }
258 public int scaleInsectTotal(int vals){
259     totalScale = 0;
260     int tooMuch = 0;
261     for (int y = 0; y <= vals; y++) {
262         totalScale += scaleInsect [y];
263         if (totalScale > 5 || totalScale < 0) {
264             tooMuch = 1;
265             break;
266         }
267     }
268     if (tooMuch == 0) {
269         return totalScale;
270     } else {
271         return -1;
272     }
273 }
274 public void addAction(int selectEvent){
275     if (currentAction == actionCount) {
276         actionCount++;
277     }
278     scenarioTemp [currentAction] = null;
279     addInsect [currentAction] = 0;
280     scaleInsect [currentAction] = 0;
281     actionValue actionDropdown.GetComponent<Dropdown>
282     ().value;//set action
283     durationValue = durationAddScenario.
284     GetComponent<InputField> ().text;//set duration
285     int cekNum;
286     somethingWrong = 0;
287     bool cekNumb = int.TryParse (durationValue, out
288     cekNum);
289     disableAllWarning ();
290     if (durationAddScenario.GetComponent<InputField>
291     ().text.Length != 0 && cekNumb) {
292         durationValue = durationAddScenario.
293         GetComponent<InputField>().text;
294         if (jumlahSeranggaDropdown.activeSelf) {//set

```



```

295         value
296         insertValue =
297         jumlahSeranggaDropdown.GetComponent<Dropdown>
298         ().value;
299         if (jumlahSeranggaDropdown.
300         GetComponent<Dropdown> ().value == 0 &&
301         addInsectTotal(actionCount) + 1 <= 10) {
302             insertValue = tambahSeranggaDropdown
303             [0].GetComponent<Dropdown> ().value;
304             addInsect[currentAction]++;
305             insertScenarioTemp (insertValue.ToString
306             ());
307             if(addInsectTotal(actionCount)==-1){
308                 somethingWrong = 1;
309                 insectWarning.SetActive (true);
310                 insectWarning.GetComponent<Text> ().text
311                 = "Jumlah Serangga Terlalu Banyak";
312             }
313         } else if
314         (jumlahSeranggaDropdown.GetComponent
315         <Dropdown>().value > 0 && addInsectTotal
316         (actionCount) + jumlahSeranggaDropdown.
317         GetComponent<Dropdown>
318         ().value + 1 <= 10) {
319             for (int p = 0; p <=
320             jumlahSeranggaDropdown.
321             GetComponent<Dropdown> ().value; p++) {
322                 insertValue = tambahSeranggaDropdown
323                 [p].GetComponent<Dropdown> ().value;
324                 addInsect[currentAction]++;
325                 if (p != jumlahSeranggaDropdown.
326                 GetComponent<Dropdown> ().value) {
327                     scenarioTemp [currentAction] +=
328                     actionValue.ToString () + "a" +
329                     insertValue.ToString () + "v" + "0d";
330                 } else if (p == jumlahSeranggaDropdown.
331                 GetComponent<Dropdown> ().value) {
332                     insertScenarioTemp (insertValue.
333                     ToString());
334                 }
335             }
336             if(addInsectTotal(actionCount)==-1){
337                 somethingWrong = 1;
338                 insectWarning.SetActive (true);
339                 insectWarning.GetComponent<Text> ().text
340                 = "Jumlah Serangga Terlalu Banyak";
341             }
342         } else {
343             somethingWrong = 1;

```

344	insectWarning.SetActive (true);
345	insectWarning.GetComponent<Text> ().text =
346	"Jumlah Serangga Terlalu Banyak";
347	}
348	} else if (kurangiSeranggaDropdown.activeSelf)
349	{
350	insertValue = kurangiSeranggaDropdown.
351	GetComponent<Dropdown> ().value + 1;
352	if (addInsectTotal(actionCount) - insertValue
353	>= 0) {
354	addInsect[currentAction] -= insertValue;
355	insertScenarioTemp (insertValue.ToString
356	());
357	if(addInsectTotal(actionCount)==-1){
358	somethingWrong = 1;
359	insectWarning.SetActive (true);
360	insectWarning.GetComponent<Text> ().text
361	= "Jumlah Serangga Terlalu Sedikit";
362	}
363	} else {
364	somethingWrong = 1;
365	insectWarning.SetActive (true);
366	insectWarning.GetComponent<Text> ().text =
367	"Jumlah Serangga Terlalu Sedikit";
368	}
369	} else if (insectSizeDropdown.activeSelf) {
370	insertValue = insectSizeDropdown.
371	GetComponent<Dropdown> ().value + 1;
372	if (scaleInsectTotal(actionCount) +
373	insertValue < 5) {
374	scaleInsect[currentAction] += insertValue;
375	insertScenarioTemp (insertValue.ToString
376	());
377	if(scaleInsectTotal(actionCount)==-1){
378	somethingWrong = 1;
379	scaleWarning.SetActive (true);
380	scaleWarning.GetComponent<Text> ().text =
381	"Ukuran Serangga Terlalu Besar";
382	}
383	} else {
384	somethingWrong = 1;
385	scaleWarning.SetActive (true);
386	scaleWarning.GetComponent<Text> ().text =
387	"Ukuran Serangga Terlalu Besar";
388	}
389	} else if (insectSizeDropdown2.activeSelf) {
390	insertValue = insectSizeDropdown2.
391	GetComponent<Dropdown> ().value + 1;
392	if (scaleInsectTotal(actionCount) -

```

393         insertValue >= 0) {
394             scaleInsect[currentAction] -= insertValue;
395             insertScenarioTemp (insertValue.ToString
396                 ());
397             if(scaleInsectTotal(actionCount)==-1){
398                 somethingWrong = 1;
399                 scaleWarning.SetActive (true);
400                 scaleWarning.GetComponent<Text> ().text =
401                     "Jumlah Serangga Terlalu Kecil";
402             }
403         }
404         else {
405             somethingWrong = 1;
406             scaleWarning.SetActive (true);
407             scaleWarning.GetComponent<Text> ().text =
408                 "Jumlah Serangga Terlalu Kecil";
409         }
410     } else {
411         insertScenarioTemp("-");
412     }
413     if (flagedit == 0) {
414         saveScenario = "scenario" + getScenarioStat
415             ().ToString ();
416     }
417     if (flagedit == 1) {
418         saveScenario = "scenario" +
419             indexedit.ToString();
420     }
421     if (selectEvent == 0 && somethingWrong==0)
422     {
423         //jika pilih save scenario
424         flushAddedScenarioTemp ();
425         PlayerPrefs.SetString (saveScenario,
426             allscenario);
427         //save scenario1, scenario2 dst
428         resetScenario ();
429         backAddScenarioButtonAction ();
430     }
431     else if (selectEvent==1 && somethingWrong==0)
432     {
433         //jika pilih tambah aksi
434         currentAction++;
435         cekActionIndex ();
436         int currentActionPrint = currentAction + 1;
437         actionLabelAddScenario.GetComponent<Text>
438             ().text = "AKSI " + currentActionPrint;
439         defaultformaddscenario ();
440     }
441     } else if (durationAddScenario.GetComponent
442         <InputField>().text.Length == 0) {
443         dataBelumLengkap.SetActive (true);
444     } else if (!cekNumb) {

```

```

442         dataWrong.SetActive (true);
443     }
444     if (somethingWrong == 0) {
445         jmlSeranggaValue.GetComponent<Text> ().text =
446         addInsectTotal (actionCount).ToString () +
447         "/10";
448         int insectsizee = scaleInsectTotal
449         (actionCount) + 1;
450         scaleValue.GetComponent<Text> ().text =
451         insectsizee.ToString () + "/5";
452     }
453 }
454 public int getScenarioStat(){
455     int j=0;
456     while(j<10){
457         string scenariokey = "scenario" + j.ToString
458         ();
459         if (PlayerPrefs.GetString (scenariokey).Length
460         == 0) {
461             break;
462         }
463         j++;
464     }
465     return j;
466 }
467 public void loadAction(){
468     for(int q=0;q<actionLength;q++){
469         if(action[q]!=null && value [q]!=null &&
470         duration[q]!=null){
471             actionDropdown.GetComponent<Dropdown>().value
472             = int.Parse(action[q]);
473             durationAddScenario.GetComponent<InputField>
474             ().text = duration [q].ToString ();
475             if (action [q] == "0") {
476                 addInsectDropdown.transform.GetChild
477                 (q).gameObject.SetActive (true);
478                 jumlahSeranggaDropdown.GetComponent
479                 <Dropdown>().value = q;
480                 addInsectDropdown.transform.GetChild
481                 (q).GetComponent<Dropdown> ().value =
482                 int.Parse
483                 (value [q].ToString ());
484             }
485             else if (action [q] == "1") {
486                 kurangiSeranggaDropdown.gameObject.
487                 SetActive(true);
488                 kurangiSeranggaDropdown.GetComponent
489                 <Dropdown>().value = int.Parse (value
490                 [q].ToString ()) - 1;

```

```

491     }
492     else if (action [q] == "4") {
493         insectSizeDropdown.gameObject.SetActive
494         (true);
495         insectSizeDropdown.GetComponent<Dropdown>
496         ().value = int.Parse (value [q].ToString
497         ()) - 1;
498     }
499     else if (action [q] == "5") {
500         insectSizeDropdown2.gameObject.SetActive
501         (true);
502         insectSizeDropdown2.GetComponent<Dropdown>
503         ().value = int.Parse (value [q].ToString
504         ()) - 1;
505     }
506 }
507 }
508 int currentActionPrint = currentAction + 1;
509 actionLabelAddScenario.GetComponent<Text> ().text
510 = "AKSI " + currentActionPrint;
511 }
512 public void loadScenario(int
513     scenarionumber){scenarionumber--;
514     string loadkey = "scenario" + scenarionumber.
515     ToString ();
516     getEachScenario (PlayerPrefs.GetString (loadkey).
517     ToString());
518     runScenario ();
519 }
520 public void defaultformaddscenario(){
521     actionDropdown.GetComponent<Dropdown> ().value =
522     0;
523     jumlahSeranggaDropdown.GetComponent<Dropdown>
524     ().value = 0;
525     for (int k = 0; k < 10; k++) {
526         addInsectDropdown.transform.GetChild
527         (k).GetComponent<Dropdown> ().value = 0;
528     }
529     durationAddScenario.GetComponent<InputField>
530     ().text = "";
531     insectSizeDropdown.GetComponent<Dropdown>
532     ().value = 0;
533     jumlahSeranggaDropdown.SetActive (true);
534     insectSizeDropdown.SetActive (false);
535     kurangiSeranggaDropdown.SetActive (false);
536     disableAllWarning ();
537     jumlahLabel.SetActive (true);
538     addInsectDropdown.transform.GetChild
539     (0).gameObject.SetActive (true);

```

```

540     }
541     public void cekSkenarioStatus(){
542         jumlahSkenario = 0;
543         for (int o = 0; o < 10; o++) {
544             string scenariokeyy = "scenario" + o.ToString
545             ();
546             if (PlayerPrefs.GetString (scenariokeyy).Length
547             > 0) {
548                 jumlahSkenario++;
549                 scenarioview [o].SetActive (true);
550                 scenarioedit [o].SetActive (true);
551                 scenariodelete [o].SetActive (true);
552             }
553             else if (PlayerPrefs.GetString (scenariokeyy).
554             Length == 0) {
555                 scenarioview [o].SetActive (false);
556                 scenarioedit [o].SetActive (false);
557                 scenariodelete [o].SetActive (false);
558             }
559         }
560         if (jumlahSkenario == 10) {
561             addScenarioButton.SetActive (false);
562         }
563         else{
564             addScenarioButton.SetActive (true);
565         }
566     }
567     public void disableAllWarning(){
568         insectWarning.SetActive (false);
569         dataBelumLengkap.SetActive (false);
570         dataWrong.SetActive (false);
571         scaleWarning.SetActive (false);
572     }
573     public void deleteScenario(int indexHapus2){
574         string deletekey = "scenario" +
575         indexHapus2.ToString ();
576         PlayerPrefs.SetString (deletekey, "");
577         PlayerPrefs.DeleteKey (deletekey);
578         cekSkenarioStatus ();
579     }
580     //-----
581     -----ButtonAction-----
582     -----//
583     public void scenarioButtonAction(){
584         scenarioContent.SetActive (true);
585         settingContent.SetActive (false);
586         cekSkenarioStatus ();
587         bgImage.SetActive (true);
588         if (isPasien == 1) {

```

```

589         addScenarioButton.SetActive (false);
590         ubahScenarioButton.SetActive (false);
591         hapusScenarioButton.SetActive (false);
592     } else {
593         addScenarioButton.SetActive (true);
594         ubahScenarioButton.SetActive (true);
595         hapusScenarioButton.SetActive (true);
596     }
597 }
598 public void backScenarioButtonAction(){
599     scenarioContent.SetActive (false);
600     if (isPasien == 1) {//jika pasien
601         mainMenuContent.SetActive (true);
602     } else {//jika terapis
603         terapisContent.SetActive (true);
604     }
605 }
606 public void addScenarioButtonAction(){
607     addScenarioContent.SetActive (true);
608     scenarioContent.SetActive (false);
609     prevActionButton.SetActive (false);
610     nextActionButton.SetActive (false);
611     disableAllWarning ();
612     int scenarionumb = getScenarioStat () + 1;
613     addScenarioLabel.GetComponent<Text> ().text =
614     "Tambah Skenario " + scenarionumb.ToString();
615     flagedit = 0;
616     defaultformaddscenario ();
617     cekActionIndex ();
618 }
619 public void backAddScenarioButtonAction(){
620     flushAddedScenarioTemp ();
621     resetScenario ();
622     if (flagedit == 0) {
623         scenarioContent.SetActive (true);
624         addScenarioContent.SetActive (false);
625         cekSkenarioStatus ();
626     }
627     else if (flagedit==1){
628         ubahSkenarioContent.SetActive (true);
629         addScenarioContent.SetActive (false);
630     }
631 }
632 public void prevActionButtonAction(){
633     addAction (2);
634     if (somethingWrong == 0) {
635         disableAllWarning ();
636         currentAction--;
637         cekActionIndex ();

```

```

638         getEachScenario (scenarioTemp [currentAction]);
639         loadAction ();
640     }
641 }
642 public void nextActionButtonAction(){
643     addAction (3);
644     if (somethingWrong == 0) {
645         disableAllWarning ();
646         if (scenarioTemp [currentAction + 1] != null) {
647             currentAction++;
648             cekActionIndex ();
649             getEachScenario (scenarioTemp
650 [currentAction]);
651             loadAction ();
652         }
653     }
654 }
655 public void cekActionIndex(){
656     prevActionButton.SetActive (false);
657     nextActionButton.SetActive (false);
658     if (currentAction == 0 && scenarioTemp [1] !=
659 null) {
660         nextActionButton.SetActive (true);
661     }
662     if (currentAction!=0 && scenarioTemp
663 [currentAction + 1] != null) {
664         nextActionButton.SetActive (true);
665         prevActionButton.SetActive (true);
666     }
667     if (currentAction!=0 && scenarioTemp
668 [currentAction - 1] != null) {
669         prevActionButton.SetActive (true);
670     }
671 }
672 public void ubahScenarioButtonAction(int
673 indexUbah){
674     ubahSkenarioContent.SetActive (true);
675     scenarioContent.SetActive (false);
676 }
677 public void backUbahScenarioButtonAction(){
678     scenarioContent.SetActive (true);
679     ubahSkenarioContent.SetActive (false);
680 }
681 public void hapusScenarioButtonAction(){
682     hapusSkenarioContent.SetActive (true);
683     scenarioContent.SetActive (false);
684     flagedit = 0;
685     cekSkenarioStatus ();
686 }

```



```

687 public void chooseDeleteScenario(int indexHapus){
688     indexHapus--;
689     deleteScenario (indexHapus);
690 }
691 public void backHapusScenarioButtonAction(){
692     scenarioContent.SetActive (true);
693     hapusSkenarioContent.SetActive (false);
694 }
695 public void editScenarioButtonAction(){
696     ubahSkenarioContent.SetActive (true);
697     ubahSkenarioContent.SetActive (false);
698     cekActionIndex ();
699     cekSkenarioStatus ();
700 }
701 public void chooseEditScenario(int indexEdit2){
702     addScenarioContent.SetActive (true);
703     ubahSkenarioContent.SetActive(false);
704     disableAllWarning ();
705     indexEdit2--;
706     string editkey = "scenario" + indexEdit2.ToString
707 ();
708     flagedit = 1;
709     indexedit = indexEdit2;
710     int indexeditprint = indexEdit2 + 1;
711     addScenarioLabel.GetComponent<Text> ().text =
712     "Ubah Skenario " + indexeditprint.ToString();
713     getEachScenario(PlayerPrefs.GetString(editkey));
714     int fill=0;
715     insertScenario = 0;
716     addInsect [insertScenario] = 0;
717     scaleInsect[insertScenario]= 0;
718     while(fill < actionLength-1){
719         scenarioTemp[insertScenario] += action[fill] +
720         "a" + value[fill] + "v" + duration[fill] +
721         "d";
722         if (action [fill] == "0") {
723             addInsect [insertScenario]++;
724         }
725         else if (action [fill] == "1") {
726             int v = int.Parse (value [fill]);
727             addInsect [insertScenario] -= v;
728         }
729         else if (action [fill] == "4") {
730             int v = int.Parse (value [fill]);
731             scaleInsect [insertScenario] = v;
732         }
733         else if (action [fill] == "5") {
734             int v = int.Parse (value [fill]);
735             scaleInsect [insertScenario] -= v;

```

```

736     }
737     if (duration[fill]!="0") {
738         insertScenario++;
739         addInsect [insertScenario] = 0;
740         scaleInsect[insertScenario] = 0;
741     }
742     fill++;
743 }
744 jmlSeranggaValue.GetComponent<Text> ().text =
745 addInsectTotal (insertScenario).ToString () +
746 "/10";
747 int insectsizee = scaleInsectTotal
748 (insertScenario) + 1;
749 scaleValue.GetComponent<Text> ().text =
750 insectsizee.ToString () + "/5";
751 scaleInsectTotal (insertScenario);
752 actionCount = insertScenario;
753 cekActionIndex ();
754 getEachScenario (scenarioTemp [0]);
755 loadAction();
756 int tempvals = jumlahSeranggaDropdown.
757 GetComponent<Dropdown> ().value;
758 jumlahSeranggaDropdown.GetComponent<Dropdown>
759 ().value = jumlahSeranggaDropdown.
760 GetComponent<Dropdown> ().value + 1;
761 jumlahSeranggaDropdown.GetComponent<Dropdown>
762 ().value = tempvals;
763 }
764 public void flushAllScenarioTemp(){
765     for (int d = 0; d < 25; d++) {
766         scenarioTemp [d] = null;
767     }
768 }
769 public void flushAddedScenarioTemp(){
770     for (int p = 0; p < actionCount; p++) {
771         allscenario += scenarioTemp [p];
772         addInsect [p] = 0;
773         scaleInsect [p] = 0;
774         scenarioTemp [p] = null;//reset isi scenario
775     }
776 }
777 public void resetScenario(){
778     actionCount = 0;
779     allscenario = "";
780     currentAction = 0;
781     int currentActionPrint = currentAction + 1;
782     actionLabelAddScenario.GetComponent<Text> ().text
783     = "AKSI " + currentActionPrint;
784     jmlSeranggaValue.GetComponent<Text> ().text =

```

```

785         "0/10";
786         scaleValue.GetComponent<Text> ().text = "1/5";
787     }
788     public void backEditScenarioButtonAction(){
789         flushAllScenarioTemp ();
790         resetScenario ();
791         scenarioContent.SetActive (true);
792         ubahSkenarioContent.SetActive(false);
793     }
794     public void pasienMenu(){
795         mainMenuContent.SetActive (false);
796         isPasien = 1;
797         settingButton.SetActive (false);
798         scenarioButtonAction ();
799     }
800     public void kelolaSkenario(){
801         scenarioButtonAction ();
802         terapisContent.SetActive (false);
803     }
804     public void terapiManual(){
805         mainMenuContent.SetActive (false);
806         bgImage.SetActive (false);
807         settingButton.SetActive (true);
808     }
809     public void terapisMenu(){
810         terapisContent.SetActive (true);
811         menuContent.SetActive (false);
812         isPasien = 0;
813     }
814     public void backToMainMenu(){
815         mainMenuContent.SetActive (true);
816         settingButton.SetActive (true);
817         bgImage.SetActive (true);
818         settingContent.SetActive (false);
819     }
820     public void backTerapisMenu(){
821         terapisContent.SetActive (false);
822         menuContent.SetActive (true);
823     }
824     public void showCreditContent(){
825         creditContent.SetActive (true);
826         creditContent.transform.GetChild (1).gameObject.
827         GetComponent<ScrollRect> ().
828         verticalNormalizedPosition = 1;
829     }
830     public void backCreditContent(){
831         creditContent.SetActive (false);
832     }
833     public void showTutorialContent(){

```

834	tutorialContent.SetActive (true);
835	}
836	public void backTurotialContent(){
837	tutorialContent.SetActive (false);
838	}
839	}

Gambar A.4 Script scenarioManager

Tabel A.1 Data responden (1)

Nama	Umur	Pekerjaan
Eno Aji	25	Pengusaha
Donny Andrean	18	Musisi
Azzam Nasrul	24	Mahasiswa
Muhammad Yusron Ramadhan	24	Mahasiswa
Yusandi Rezki Fadli	23	Mahasiswa S2
Hermi	23	Mahasiswa
Muhammad Ali Anwar	26	Mahasiswa S2
Purna Hadi Swasono	20	Mahasiswa
Randytia	25	Mahasiswa
Oki	28	Event Organizer
Faris Muhammad Asmawi	24	Mahasiswa
Rahmad Cahyo Gumilar	23	Karyawan Swasta
Panji Andrean sundoro	24	Karyawan Swasta
Fajar Ainur Rofiq	27	Pengusaha
Ahmad Fauzi	29	Karyawan Swasta
Muhammad Khomarudin Bahar	20	Mahasiswa
Jordy	24	Mahasiswa
Eksels Van	17	Pelajar
Leadistia	18	Pelajar
Nisa	19	Mahasiswa

Tabel A.2 Data responden (2)

Nama	Serangga Yang Ditakuti	Sejak Kapan Takut Serangga	Kenapa Takut Serangga
Eno Aji	Kecoa, Laba-laba	Sejak lahir	Jijik
Donny Andrean	Laba-laba	SD	Takut bulunya
Azzam Nasrul	Belalang, Laba-laba, Kecoa	Sejak lahir	Takut digigit
Muhammad Yusron Ramadhan	Kecoa	Sejak kecil	Jijik dan bau
Yusandi Rezki Fadli	Kecoa	Dari S1	Takut kena kuman penyakit
Hermi	Kecoa, Kelabang	Dari sd	Geli
Muhammad Ali Anwar	Kecoa	SD	Pernah kena kencing kecoa
Purna Hadi Swasono	Kecoa	SMA	Takut kena penyakit
Randytia	Kecoa, Laba laba, Tarantula	Sejak kecil	Jijik

Oki	Kecoa	Dari kecil	Jorok dan geli
Faris Muhammad Asmawi	Kecoa, Kalajengking, Belalang	SD	jijik dan menakutkan
Rahmad Cahyo Gumilar	Kecoa	Dari smp	jijik
Panji Andrean sundoro	Kecoa, Laba-laba, lebah	Dari kecil	dari kecil, takut diserang
Fajar Ainur Rofiq	Kecoa	Dari kecil	jijik
Ahmad Fauzi	Kecoa	Dari kecil	geli
Muhammad Khomarudin Bahar	Kecoa	Dari 5 tahun	jijik dan geli
Jordy	Laba-laba, Kecoa	SD	jijik
Eksels Van	Kecoa	Dari kecil	geli
Leadistia	Kecoa	Dari kecil	geli
Nisa	Kecoa, Laba-laba	Dari kecil	jijik

Tabel A.3 Data responden (3)

Nama	Skala Takut Serangga	Skala Ingin Terapi
Eno Aji	9	10
Donny Andrean	7	5
Azzam Nasrul	8	6
Muhammad Yusron Ramadhan	7	9
Yusandi Rezki Fadli	8	10
Hermi	7	8
Muhammad Ali Anwar	8	10
Purna Hadi Swasono	6	5
Randytia	8	10
Oki	8	8
Faris Muhammad Asmawi	6	7
Rahmad Cahyo Gumilar	8	7
Panji Andrean sundoro	9	10
Fajar Ainur Rofiq	10	5
Ahmad Fauzi	7	5
Muhammad Khomarudin Bahar	7	8
Jordy	6	5

Eksels Van	5	5
Leadistia	5	5
Nisa	10	6

Tabel A.4 Data terapi (1)

Nama	Serangga Kecil			
	1 Diam	1 Jalan	Banyak Diam	Banyak Jalan
Eno Aji	7	8	8	6
Donny Andrean	5	7	5	8
Azzam Nasrul	7	7	7	7
Muhammad Yusron Ramadhan	6	7	8	8
Yusandi Rezki Fadli	5	5	7	8
Hermi	3	7	4	6
Muhammad Ali Anwar	8	10	8	10
Purna Hadi Swasono	6	7	7	7
Randytia	7	6	10	10
Oki	8	8	8	8
Faris Muhammad Asmawi	8	9	9	10
Rahmad Cahyo Gumilar	7	7	9	10
Panji Andrean sundoro	8	8	8	9
Fajar Ainur Rofiq	3	6	6	7
Ahmad Fauzi	4	5	5	5
Muhammad Khomarudin Bahar	5	7	7	8
Jordy	7	8	7	8

Eksels Van	5	7	9	10
Leadistia	5	8	10	10
Nisa	5	7	10	10

Tabel A.5 Data terapi (2)

Nama	Serangga Besar			
	1 Diam	1 Jalan	Banyak Diam	Banyak Jalan
Eno Aji	6	6	5	5
Donny Andrean	6	7	6	8
Azzam Nasrul	8	8	8	9
Muhammad Yusron Ramadhan	6	8	9	9
Yusandi Rezki Fadli	5	6	9	10
Hermi	3	6	2	5
Muhammad Ali Anwar	7	8	8	9
Purna Hadi Swasono	6	7	7	7
Randytia	5	9	10	10
Oki	7	7	7	7
Faris Muhammad Asmawi	8	10	9	10
Rahmad Cahyo Gumilar	9	10	10	10
Panji Andrean sundoro	10	10	9	10
Fajar Ainur Rofiq	6	6	7	7
Ahmad Fauzi	5	5	5	5
Muhammad Khomarudin Bahar	5	5	7	9
Jordy	8	8	8	8

Eksels Van	5	5	10	10
Leadistia	5	7	10	10
Nisa	5	8	10	10

Tabel A.6 Data terapi (3)

Nama	Membunuh Serangga				Rata-Rata Data Terapi (1)(2)(3)
	1 Diam	1 Jalan	Banyak Diam	Banyak Jalan	
Eno Aji	4	4	3	3	5,42
Donny Andrean	5	7	5	8	6,42
Azzam Nasrul	8	9	6	5	7,42
Muhammad Yusron Ramadhan	8	4	6	6	7,08
Yusandi Rezki Fadli	7	8	8	5	6,92
Hermi	4	2	1	2	3,75
Muhammad Ali Anwar	8	8	8	7	8,25
Purna Hadi Swasono	8	8	7	8	7,08
Randytia	4	4	5	5	7,08
Oki	5	5	7	7	7,00
Faris Muhammad Asmawi	6	6	6	5	8,00
Rahmad Cahyo Gumilar	10	10	10	10	9,33
Panji Andrean sundoro	10	7	10	10	9,08

Fajar Ainur Rofiq	10	10	10	10	7,33
Ahmad Fauzi	4	4	4	3	4,50
Muhammad Khomarudin Bahar	7	4	6	8	6,50
Jordy	7	7	7	7	7,50
Eksels Van	10	10	10	10	8,42
Leadistia	7	7	10	10	8,25
Nisa	5	7	9	10	8,00

Tabel A.7 Data terapi uji coba *immersive*

Nama	Hadir Dalam Situasi	Bersama Serangga	Seberapa Nyata Serangga	Rata-Rata
Eno Aji	8	8	7	7,67
Donny Andrean	7	9	6	7,33
Azzam Nasrul	7	6	5	6,00
Muhammad Yusron Ramadhan	9	9	8	8,67
Yusandi Rezki Fadli	9	9	10	9,33
Hermi	7	7	7	7,00
Muhammad Ali Anwar	6	8	9	7,67
Purna Hadi Swasono	5	5	8	6,00
Randytia	4	8	8	6,67
Oki	8	7	8	7,67
Faris Muhammad Asmawi	8	8	10	8,67
Rahmad Cahyo Gumilar	10	8	8	8,67
Panji Andrean sundoro	9	9	9	9,00
Fajar Ainur Rofiq	7	7	6	6,67
Ahmad Fauzi	6	6	4	5,33

Muhammad Khomarudin Bahar	7	8	7	7,33
Jordy	6	6	7	6,33
Eksels Van	6	7	5	6,00
Leadistia	8	8	5	7,00
Nisa	7	7	9	7,67

Tabel A.8 Data kepuasan terapi

Responden	Nilai Kepuasan
Eno Aji	10
Donny Andrean	7
Azzam Nasrul	7
Muhammad Yusron Ramadhan	8
Yusandi Rezki Fadli	9
Hermi	7
Muhammad Ali Anwar	8
Purna Hadi Swasono	8
Randytia	8
Oki	8
Faris Muhammad Asmawi	9
Rahmad Cahyo Gumilar	9
Panji Andrean sundoro	7
Fajar Ainur Rofiq	7
Ahmad Fauzi	7
Muhammad Khomarudin Bahar	8
Jordy	5
Eksels Van	6
Leadistia	10
Nisa	10

BIODATA PENULIS



Penulis lahir di Padang, 5 Maret 1993 merupakan anak pertama dari 4 bersaudara. Dalam menjalani pendidikan semasa hidup, penulis menempuh pendidikan di SDN 03 Jakata, SMPN 109 Jakarta, SMAN 71 Jakarta, dan S1 Jurusan Teknik Informatika Institut Teknologi Sepuluh Nopember (ITS).

Dalam menyelesaikan pendidikan S1 penulis mengambil bidang minat Interaksi Grafika dan Seni. Penulis memiliki ketertarikan di bidang *Game Development*, Interaksi Manusia & Komputer, *Augmented Reality*, *Virtual Reality*, dan *Web Development*.